FIG. 1

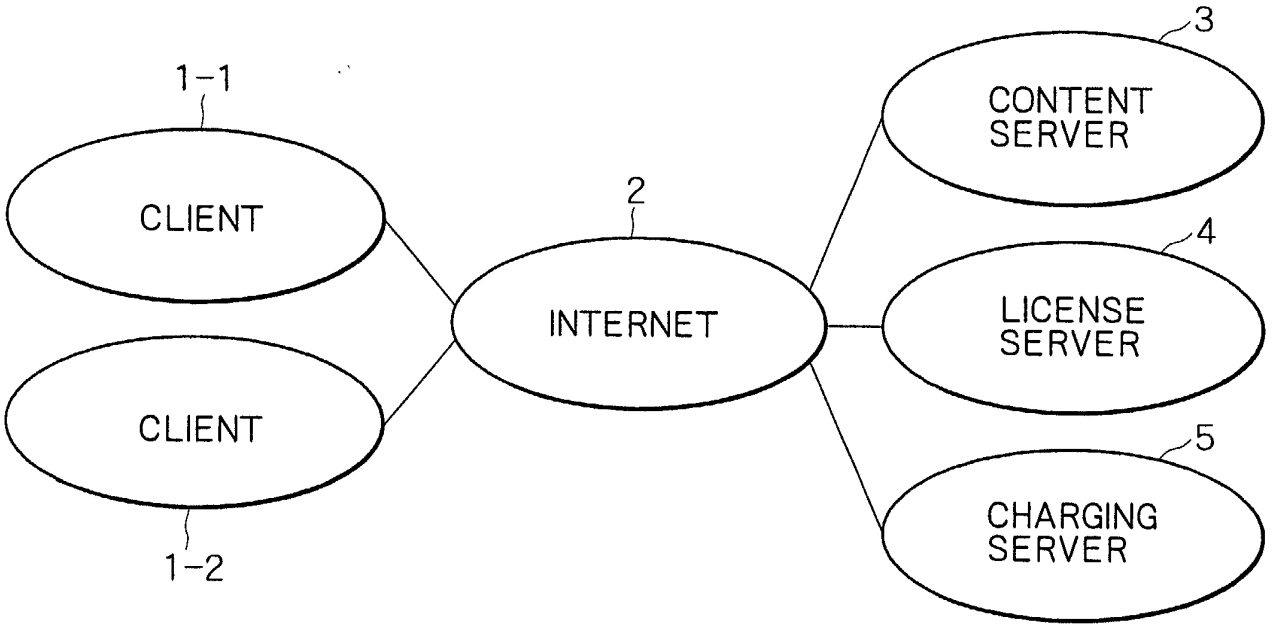


FIG. 2

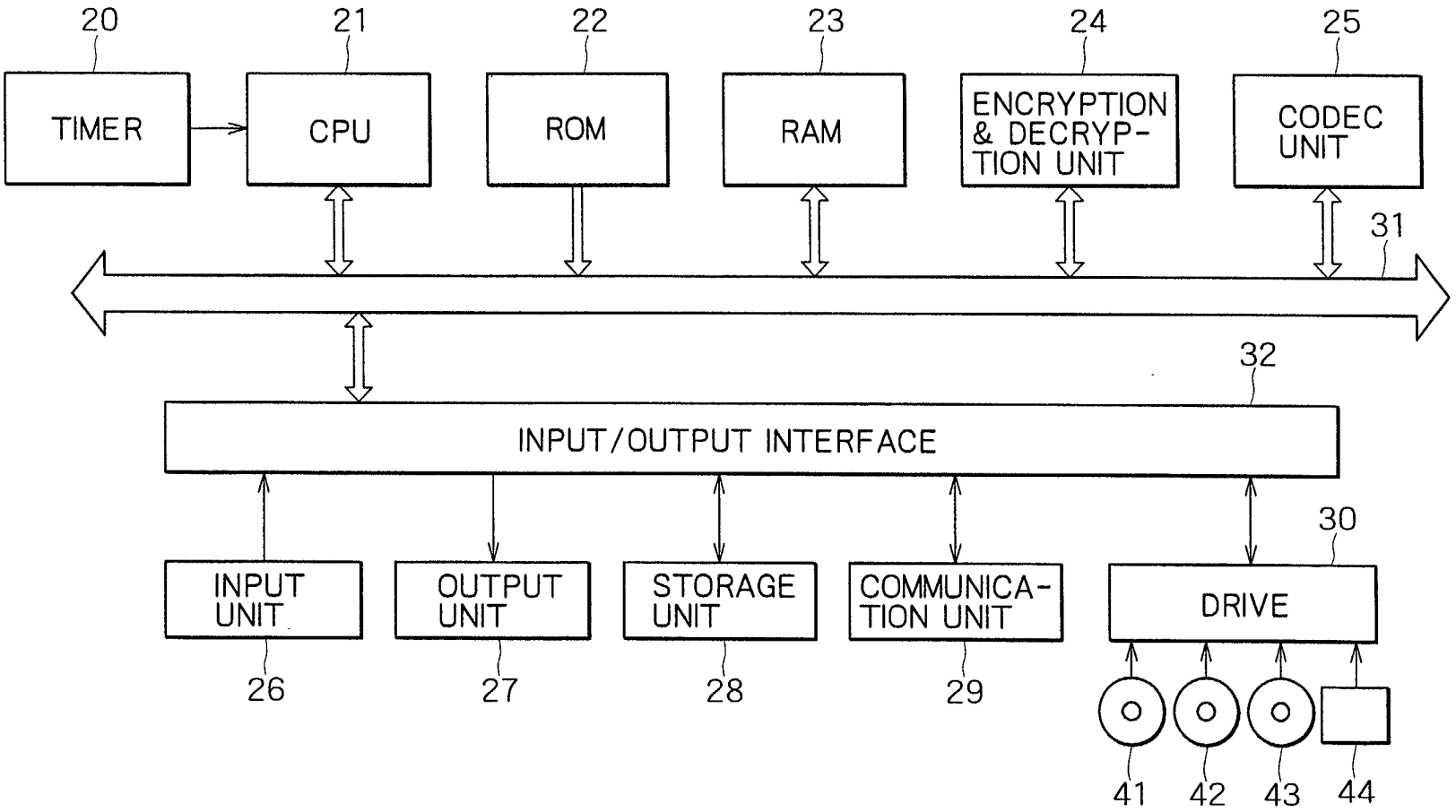

20
21
22
23
24
25
TIMER
CPU
ROM
RAM
ENCRYPTION & DECRYP-TION UNIT
CODEC UNIT
31
32
INPUT/OUTPUT INTERFACE
30
INPUT UNIT
OUTPUT UNIT
STORAGE UNIT
COMMUNICA-TION UNIT
DRIVE
26
27
28
29
41
42
43
44

FIG. 3

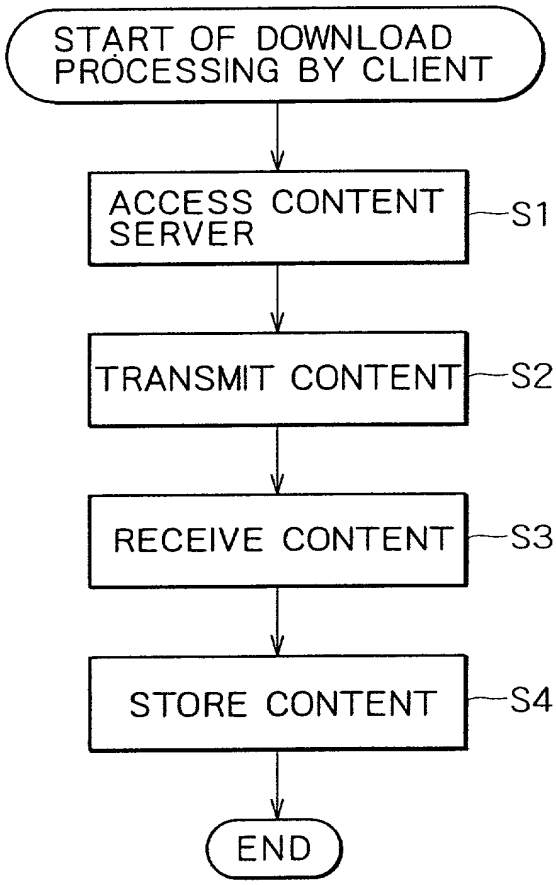

START OF DOWNLOAD PROCESSING BY CLIENT
ACCESS CONTENT SERVER
S1
TRANSMIT CONTENT
S2
RECEIVE CONTENT
S3
STORE CONTENT
S4
END

FIG. 4

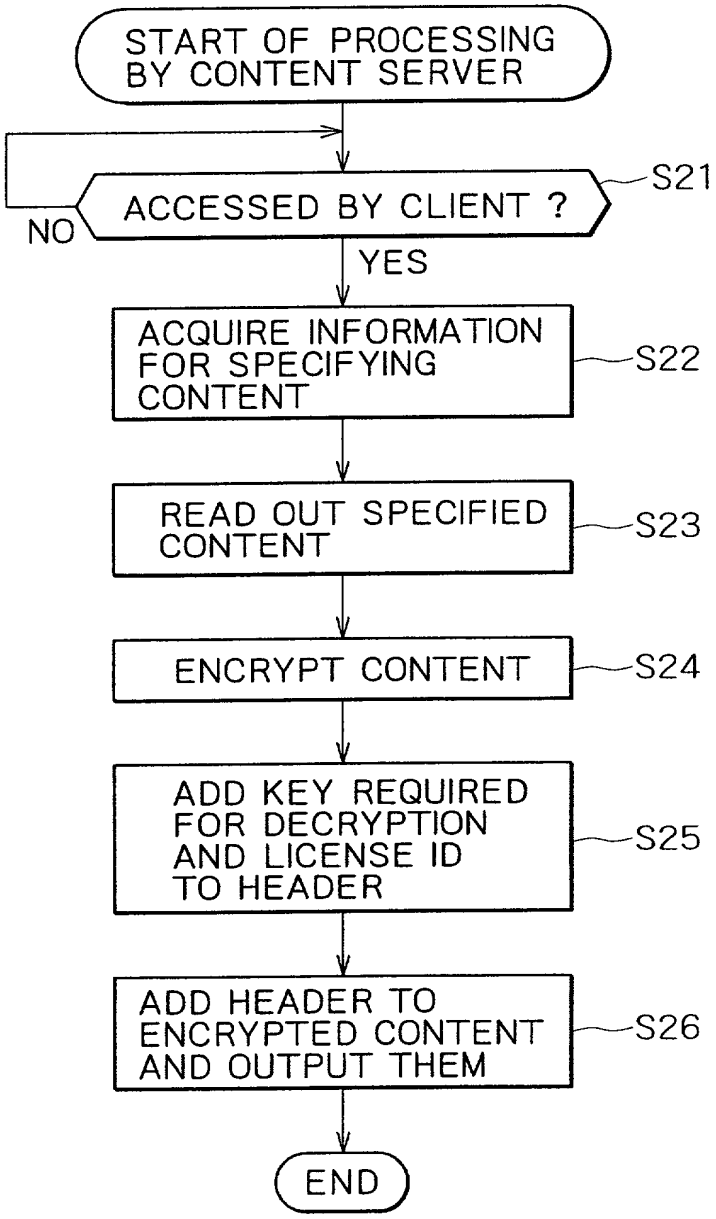

START OF PROCESSING BY CONTENT SERVER
ACCESSED BY CLIENT ?
S21
NO
YES
ACQUIRE INFORMATION FOR SPECIFYING CONTENT
S22
READ OUT SPECIFIED CONTENT
S23
ENCRYPT CONTENT
S24
ADD KEY REQUIRED FOR DECRYPTION AND LICENSE ID TO HEADER
S25
ADD HEADER TO ENCRYPTED CONTENT AND OUTPUT THEM
S26
END

FIG. 5

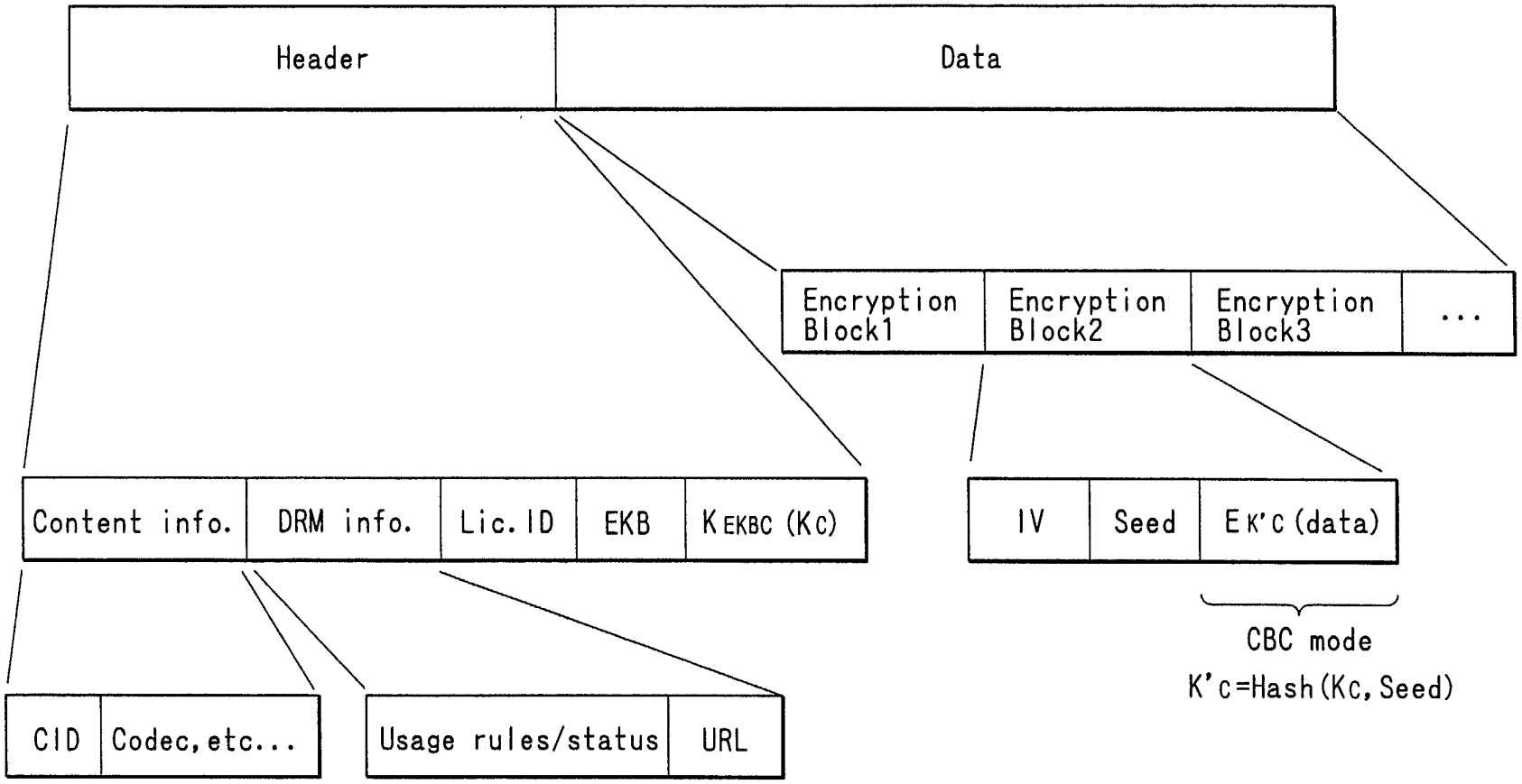

Header
Data
Encryption Block1
Encryption Block2
Encryption Block3
...
Content info.
DRM info.
Lic. ID
EKB
KEKBC (KC)
IV
Seed
EK'C (data)
CBC mode
K'C=Hash (KC, Seed)
CID
Codec, etc...
Usage rules/status
URL

FIG. 6

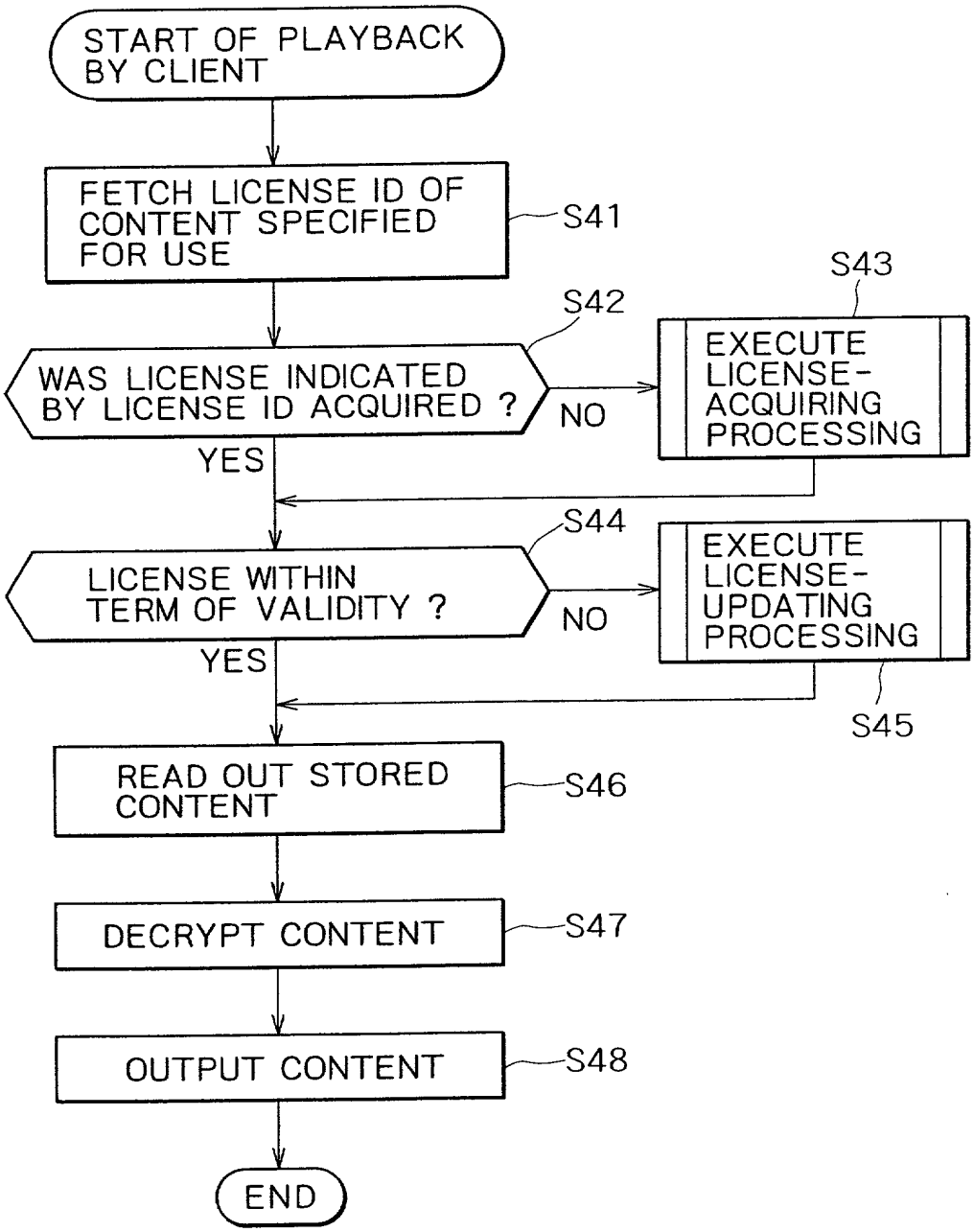

START OF PLAYBACK BY CLIENT
FETCH LICENSE ID OF CONTENT SPECIFIED FOR USE
S41
WAS LICENSE INDICATED BY LICENSE ID ACQUIRED ?
S42
NO
YES
EXECUTE LICENSE-ACQUIRING PROCESSING
S43
LICENSE WITHIN TERM OF VALIDITY ?
S44
NO
YES
EXECUTE LICENSE-UPDATING PROCESSING
S45
READ OUT STORED CONTENT
S46
DECRYPT CONTENT
S47
OUTPUT CONTENT
S48
END

FIG. 7

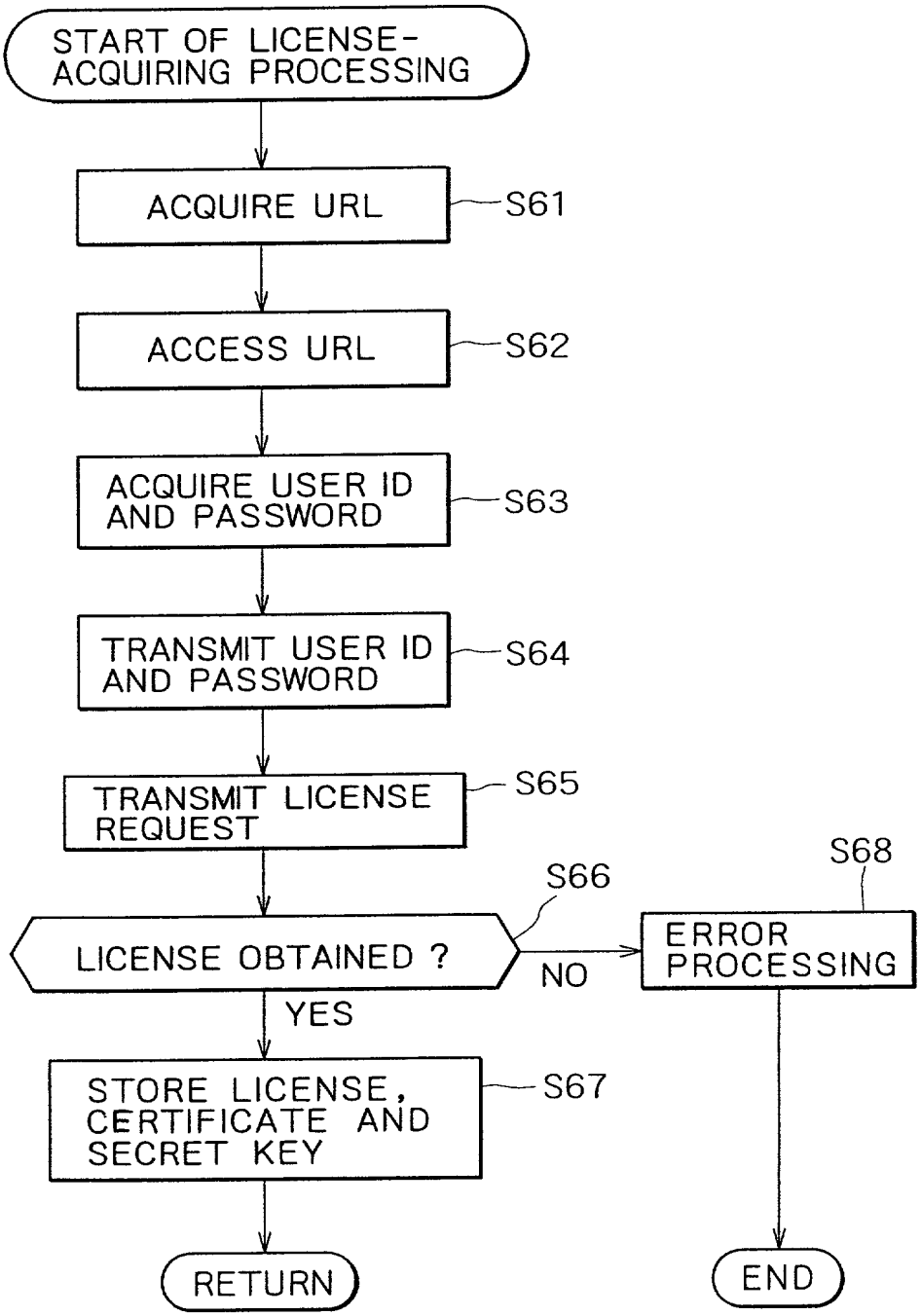

START OF LICENSE-ACQUIRING PROCESSING
ACQUIRE URL
S61
ACCESS URL
S62
ACQUIRE USER ID AND PASSWORD
S63
TRANSMIT USER ID AND PASSWORD
S64
TRANSMIT LICENSE REQUEST
S65
S66
LICENSE OBTAINED ?
NO
S68
ERROR PROCESSING
YES
STORE LICENSE, CERTIFICATE AND SECRET KEY
S67
RETURN
END

FIG.8

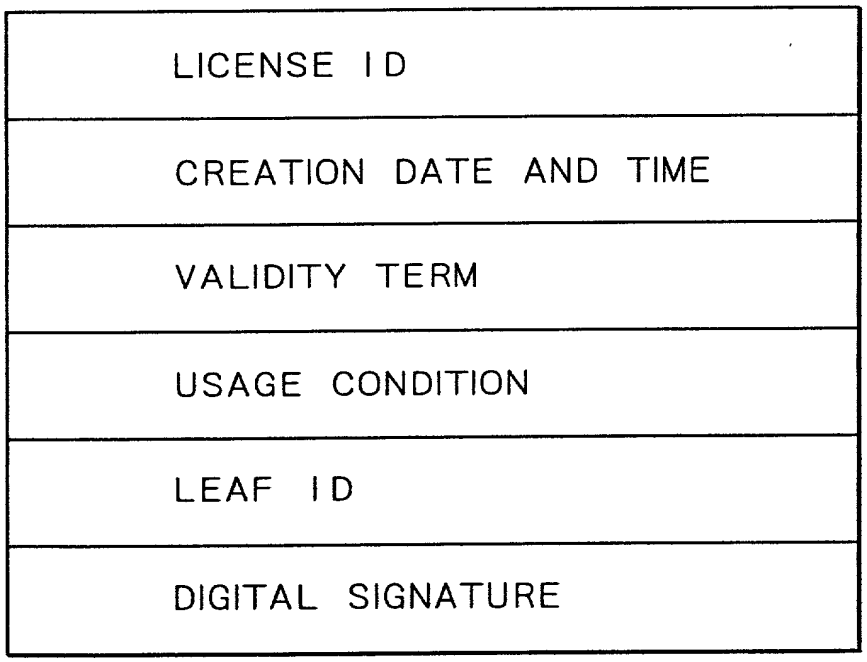

LICENSE ID
CREATION DATE AND TIME
VALIDITY TERM
USAGE CONDITION
LEAF ID
DIGITAL SIGNATURE

FIG. 9

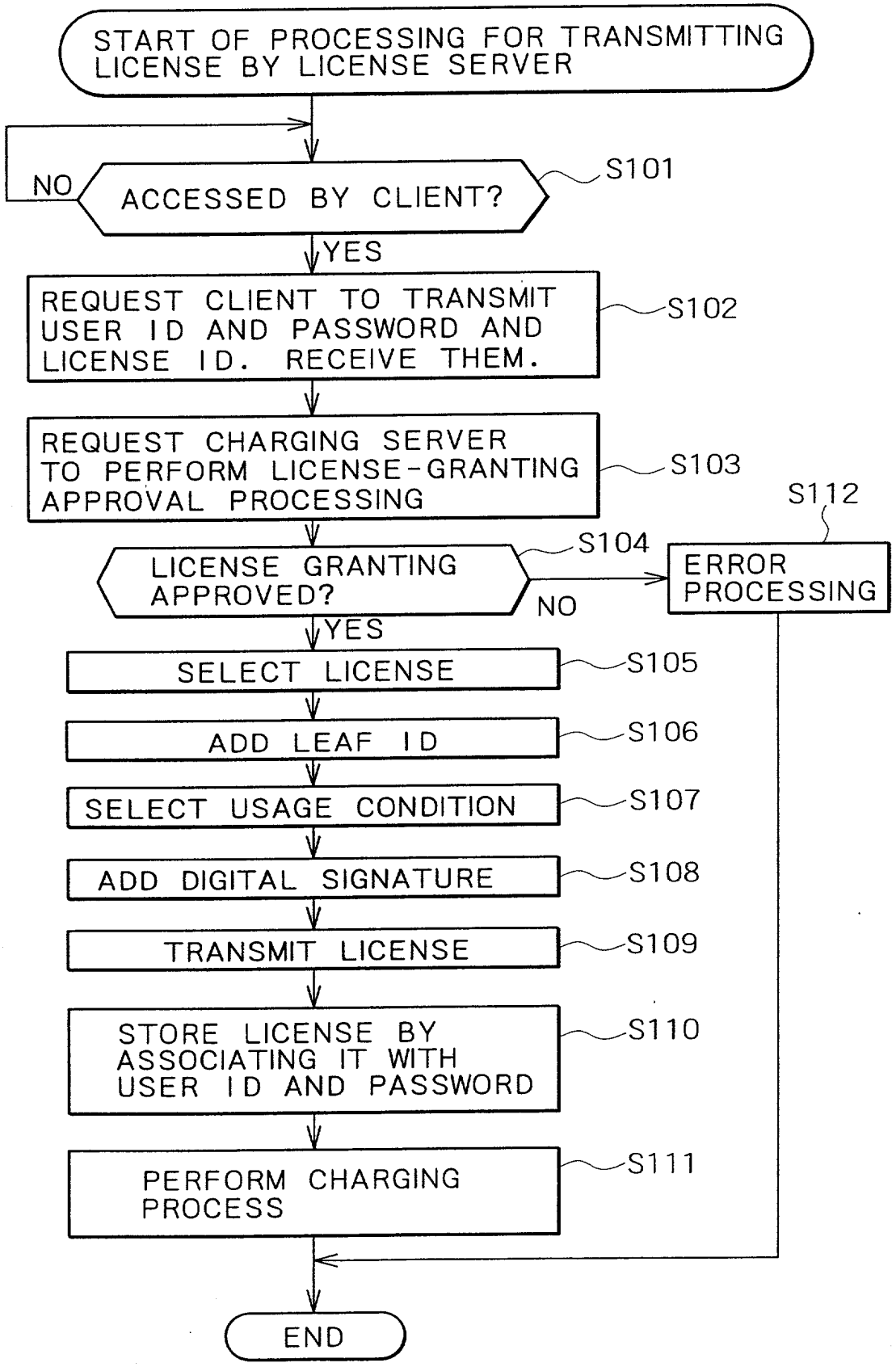

START OF PROCESSING FOR TRANSMITTING LICENSE BY LICENSE SERVER
NO
ACCESSED BY CLIENT?
S101
YES
REQUEST CLIENT TO TRANSMIT USER ID AND PASSWORD AND LICENSE ID. RECEIVE THEM.
S102
REQUEST CHARGING SERVER TO PERFORM LICENSE-GRANTING APPROVAL PROCESSING
S103
S112
S104
LICENSE GRANTING APPROVED?
NO
ERROR PROCESSING
YES
SELECT LICENSE
S105
ADD LEAF ID
S106
SELECT USAGE CONDITION
S107
ADD DIGITAL SIGNATURE
S108
TRANSMIT LICENSE
S109
STORE LICENSE BY ASSOCIATING IT WITH USER ID AND PASSWORD
S110
PERFORM CHARGING PROCESS
S111
END

FIG. 10

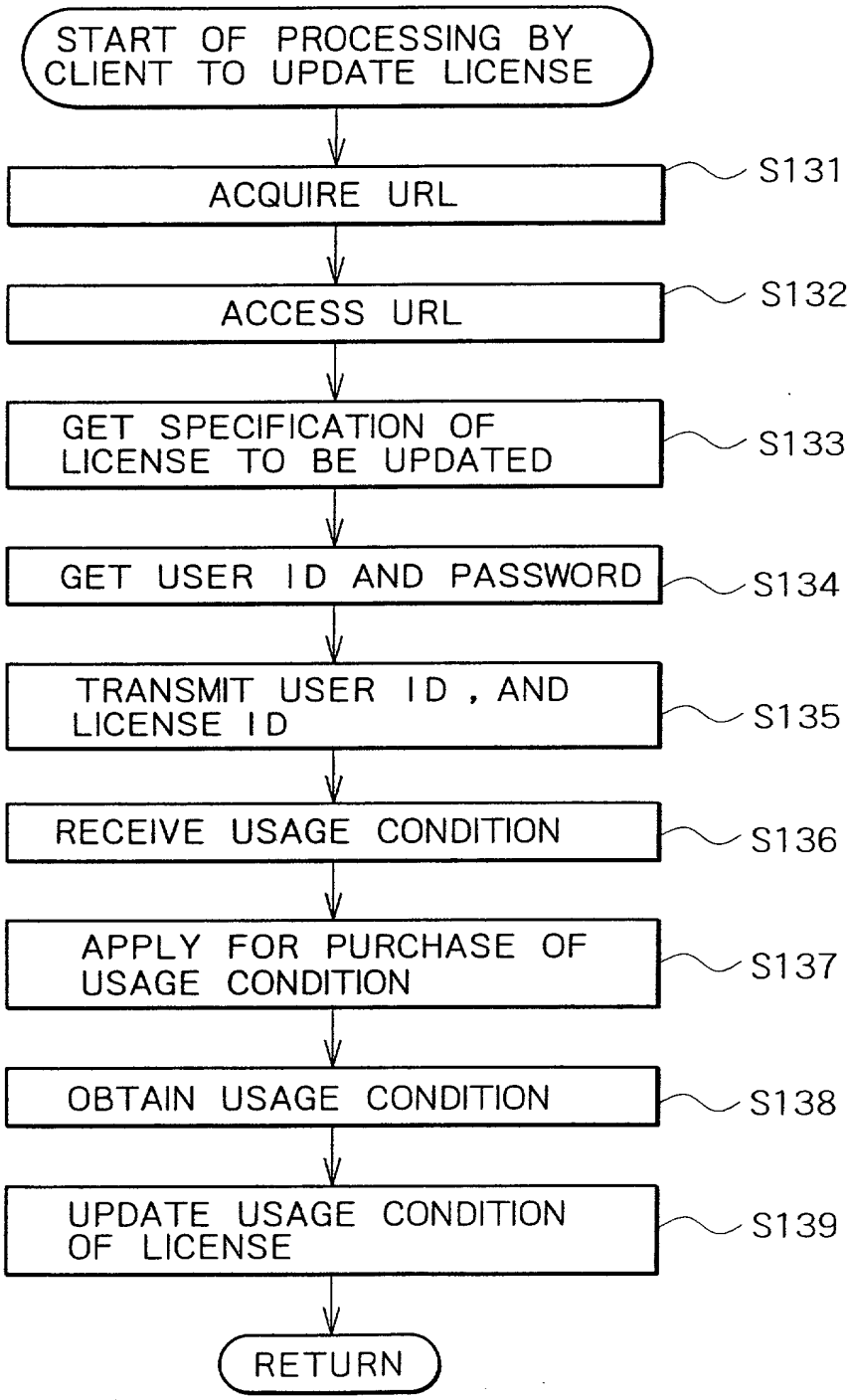

START OF PROCESSING BY CLIENT TO UPDATE LICENSE
ACQUIRE URL
S131
ACCESS URL
S132
GET SPECIFICATION OF LICENSE TO BE UPDATED
S133
GET USER ID AND PASSWORD
S134
TRANSMIT USER ID , AND LICENSE ID
S135
RECEIVE USAGE CONDITION
S136
APPLY FOR PURCHASE OF USAGE CONDITION
S137
OBTAIN USAGE CONDITION
S138
UPDATE USAGE CONDITION OF LICENSE
S139
RETURN

FIG. 11

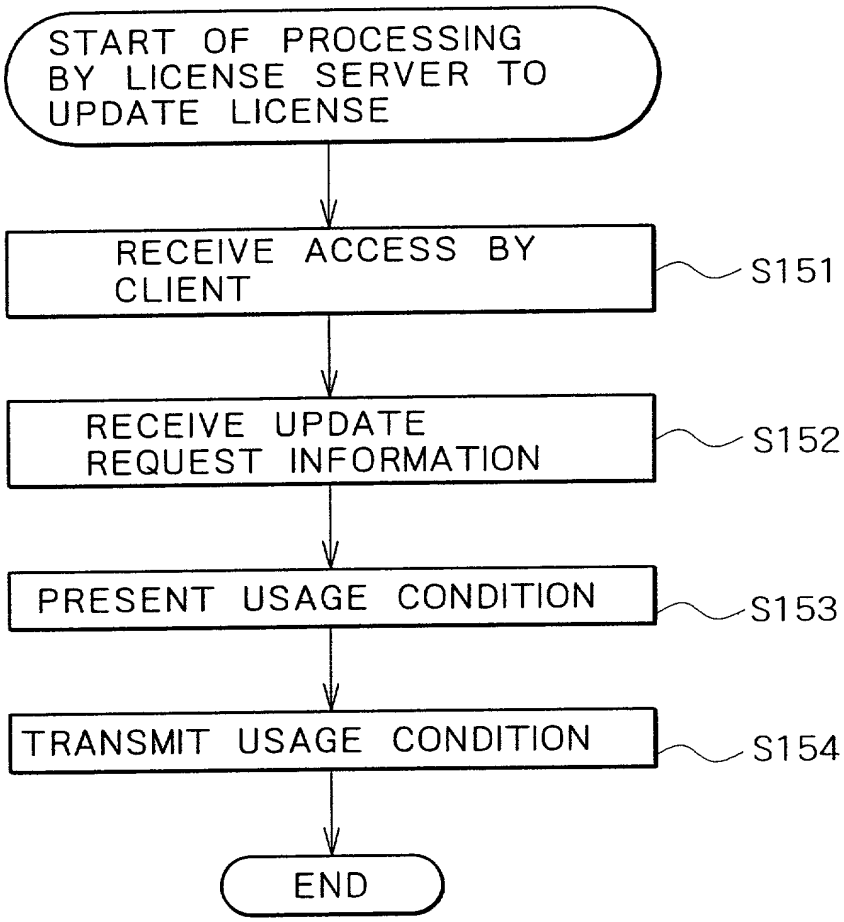

START OF PROCESSING BY LICENSE SERVER TO UPDATE LICENSE
RECEIVE ACCESS BY CLIENT
S151
RECEIVE UPDATE REQUEST INFORMATION
S152
PRESENT USAGE CONDITION
S153
TRANSMIT USAGE CONDITION
S154
END

FIG.12

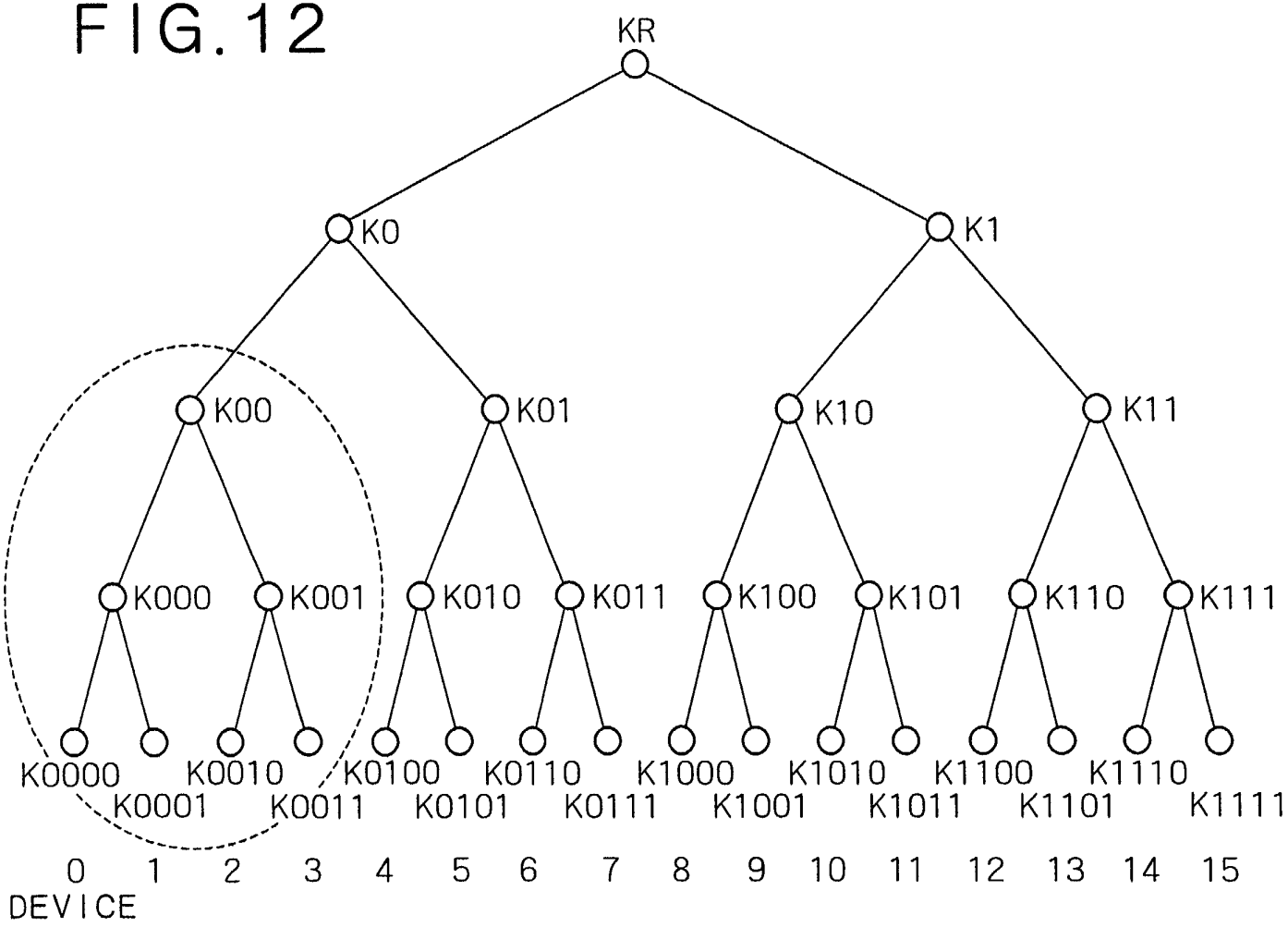

KR
K0
K1
K00
K01
K10
K11
K000
K001
K010
K011
K100
K101
K110
K111
K0000
K0001
K0010
K0011
K0100
K0101
K0110
K0111
K1000
K1001
K1010
K1011
K1100
K1101
K1110
K1111
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
DEVICE

FIG. 13

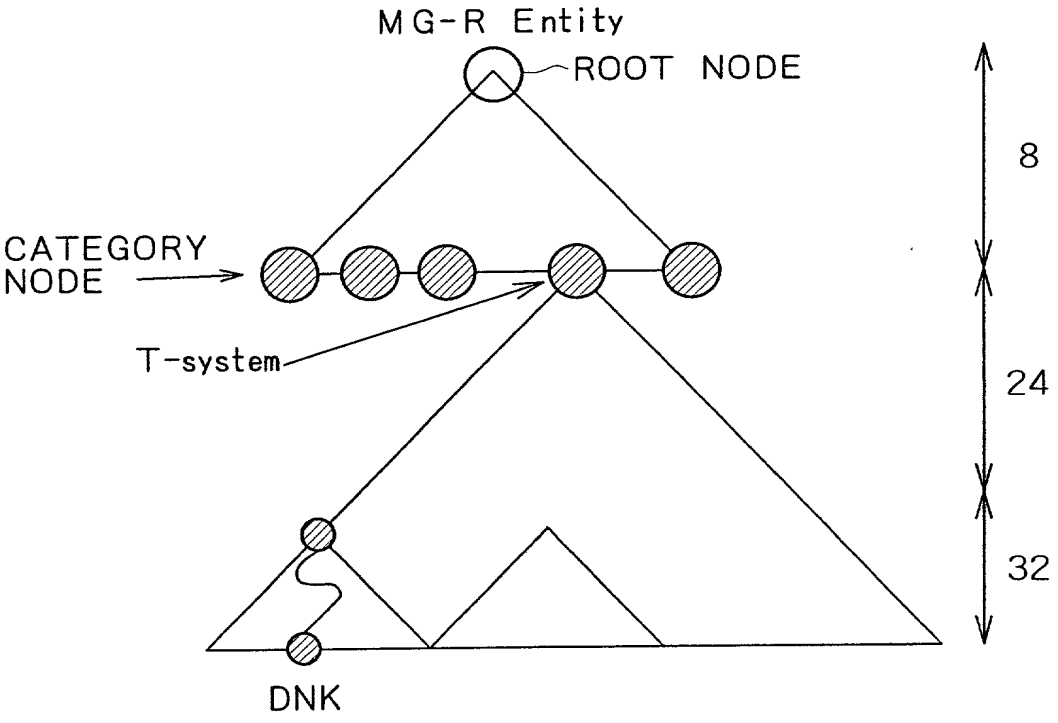


FIG. 14

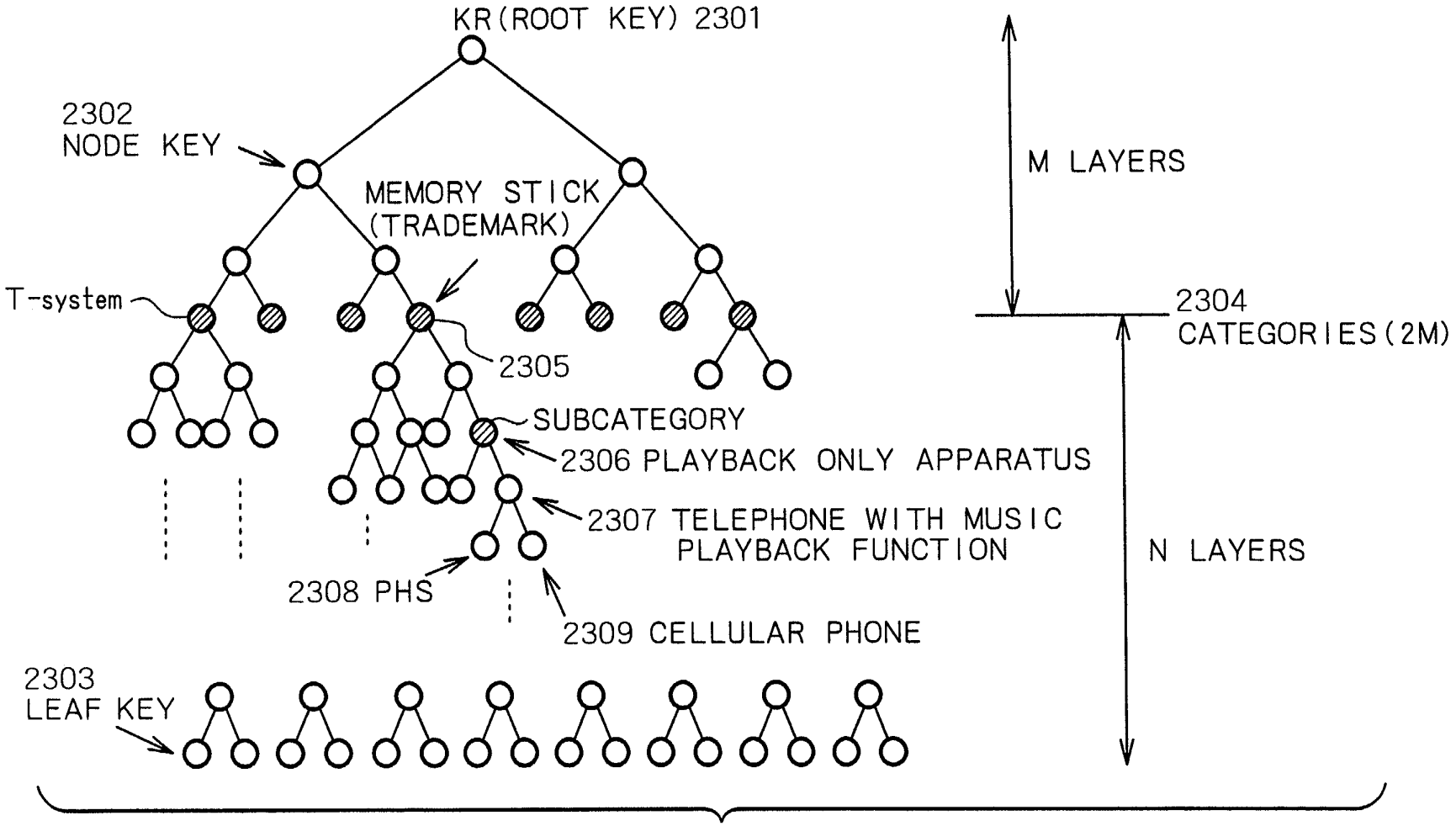

KR(ROOT KEY) 2301
2302
NODE KEY
MEMORY STICK
(TRADEMARK)
T-system
2305
SUBCATEGORY
2306 PLAYBACK ONLY APPARATUS
2307 TELEPHONE WITH MUSIC
PLAYBACK FUNCTION
2308 PHS
2309 CELLULAR PHONE
2303
LEAF KEY
M LAYERS
2304
CATEGORIES(2M)
N LAYERS

FIG. 15A

EKB (ENABLING KEY BLOCK)
TRANSMISSION OF NODE KEYS OF VERSION t TO DEVICES 0, 1 AND 2

VERSION:t	
INDEX	ENCRYPTION KEY
0	Enc(K(t)0, K(t)R)
00	Enc(K(t)00, K(t)0)
000	Enc(K000, K(t)00)
001	Enc(K(t)001, K(t)00)
0010	Enc(K0010, K(t)001)

FIG. 15B

EKB (ENABLING KEY BLOCK)
TRANSMIT NODE KEYS OF VERSION t TO DEVICES 0, 1 AND 2

VERSION:t	
INDEX	ENCRYPTION KEY
000	Enc(K000, K(t)00)
001	Enc(K(t)001, K(t)00)
0010	Enc(K0010, K(t)001)

FIG. 16

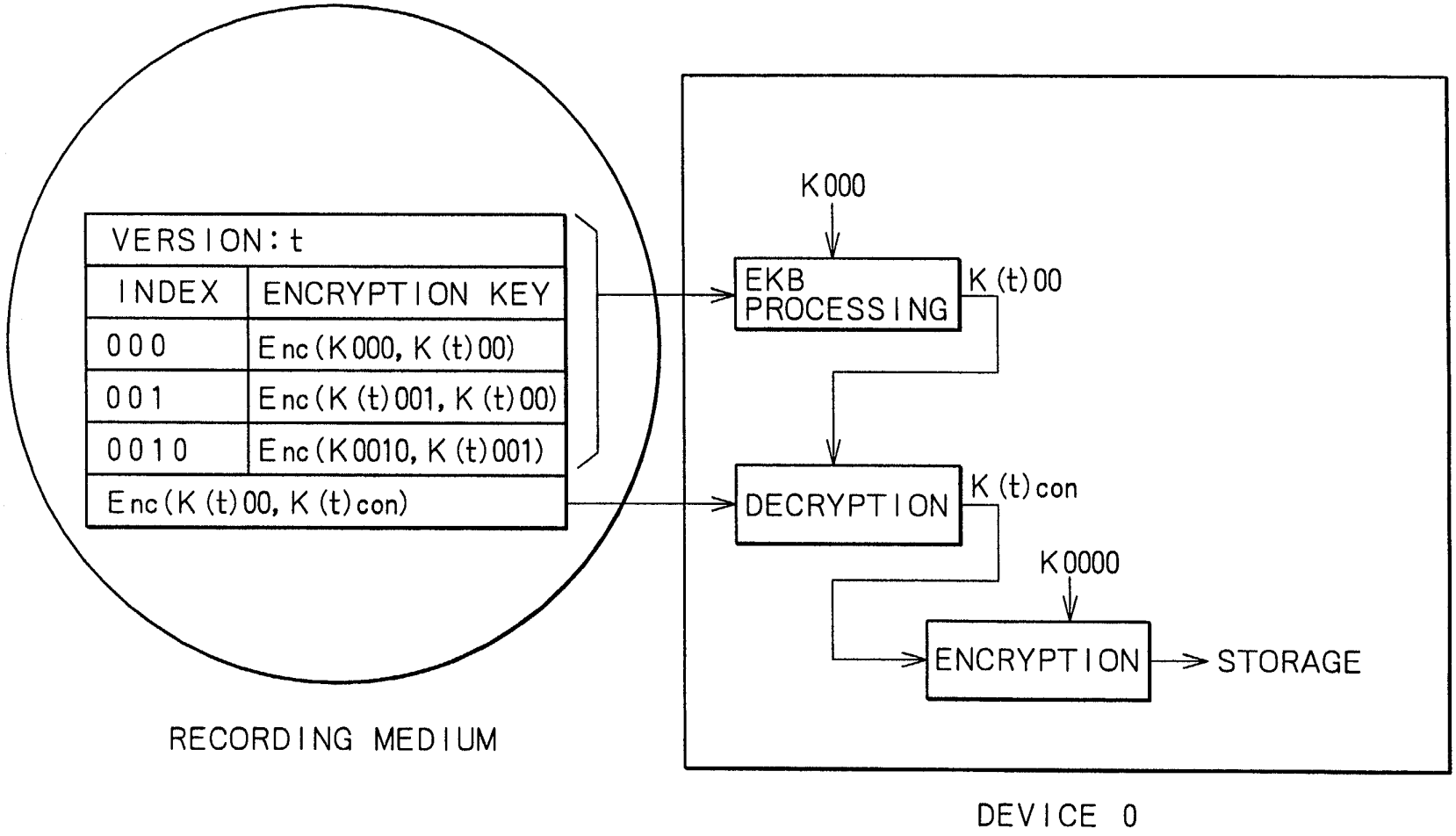

VERSION: t
INDEX
ENCRYPTION KEY
000
Enc(K000, K(t)00)
001
Enc(K(t)001, K(t)00)
0010
Enc(K0010, K(t)001)
Enc(K(t)00, K(t)con)
RECORDING MEDIUM
K000
EKB PROCESSING
K(t)00
DECRYPTION
K(t)con
K0000
ENCRYPTION
STORAGE
DEVICE 0

FIG. 17

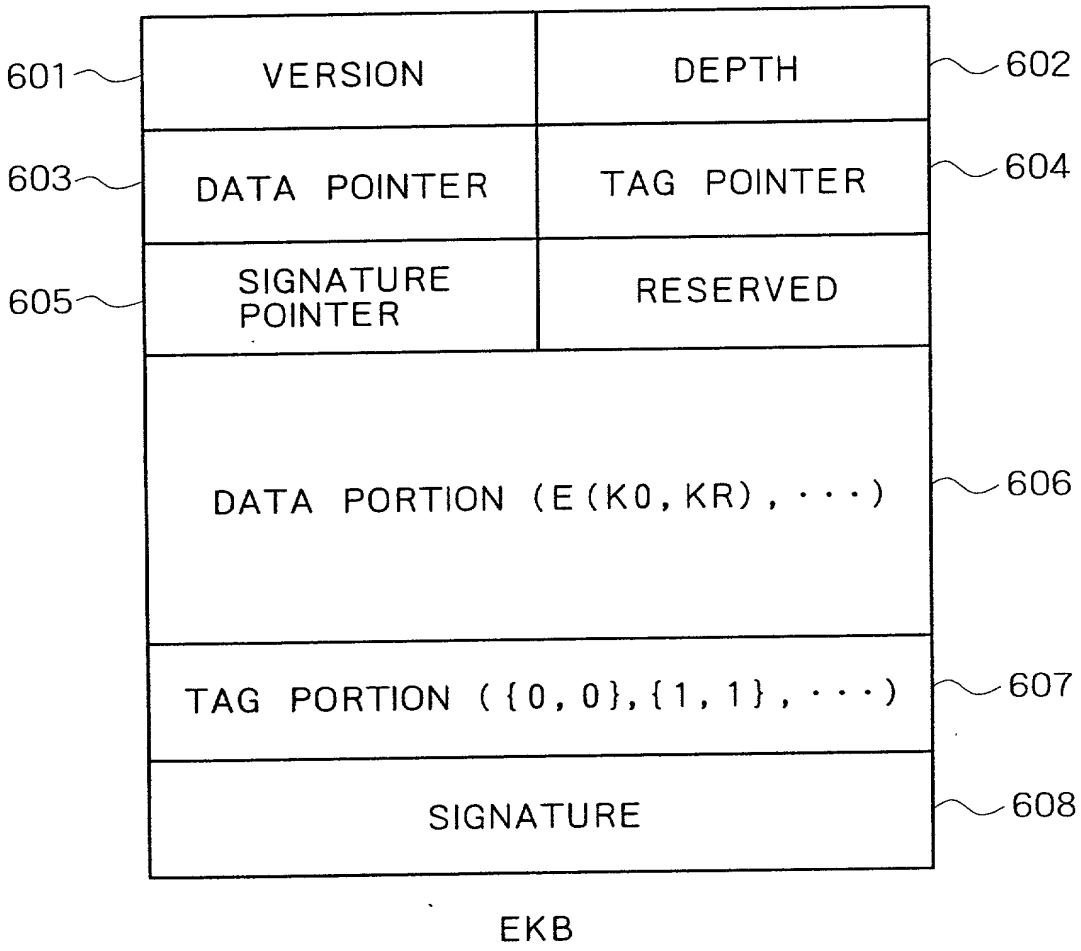

601
VERSION
DEPTH
602
603
DATA POINTER
TAG POINTER
604
605
SIGNATURE POINTER
RESERVED
DATA PORTION (E(K0, KR), ···)
606
TAG PORTION ({0, 0}, {1, 1}, ···)
607
SIGNATURE
608
EKB

FIG. 18A

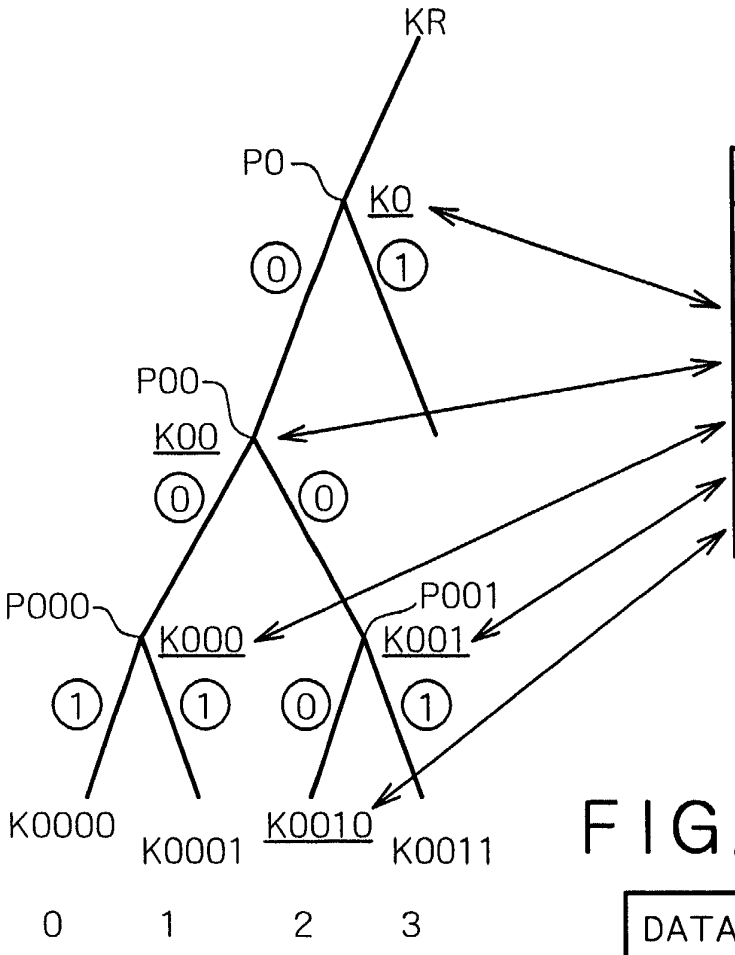


FIG. 18B

EKB (ENABLING KEY BLOCK)

TRANSMISSION OF NODE KEYS OF VERSION t TO DEVICES 0, 1 AND 2

TOP NODE ADDRESS: KR	
DATA (ENCRYPTION KEY)	TAG
Enc(K(t)0, K(t)R)	{0, 1}
Enc(K(t)00, K(t)0)	{0, 0}
Enc(K000, K(t)00)	{1, 1}
Enc(K(t)001, K(t)00)	{0, 1}
Enc(K0010, K(t)001)	{1, 1}

{L TAG, R TAG}
L TAG AND R TAG ARE EACH 0 TO INDICATE EXISTENCE OF DATA OR 1 OTHERWISE

FIG. 18C

DATA: Enc(K(t)0, K(t)R), Enc(K(t)00, K(t)0), ···
TAG: {0, 1}, {0, 0}, {1, 1} ···

FIG. 19

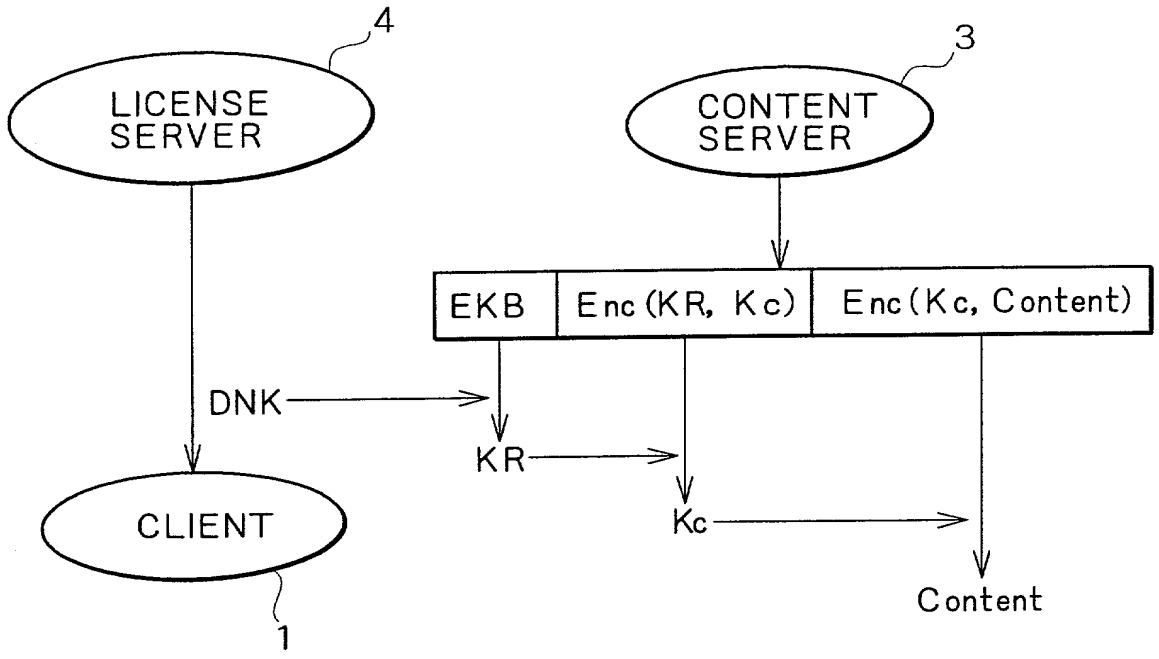

4
LICENSE SERVER
3
CONTENT SERVER
EKB
Enc(KR, Kc)
Enc(Kc, Content)
DNK
KR
Kc
Content
CLIENT
1

FIG. 20

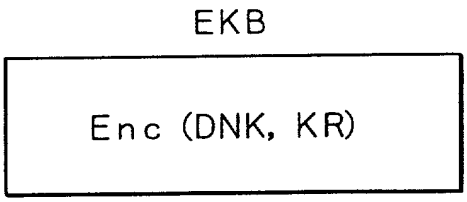

EKB
Enc (DNK, KR)

FIG. 21

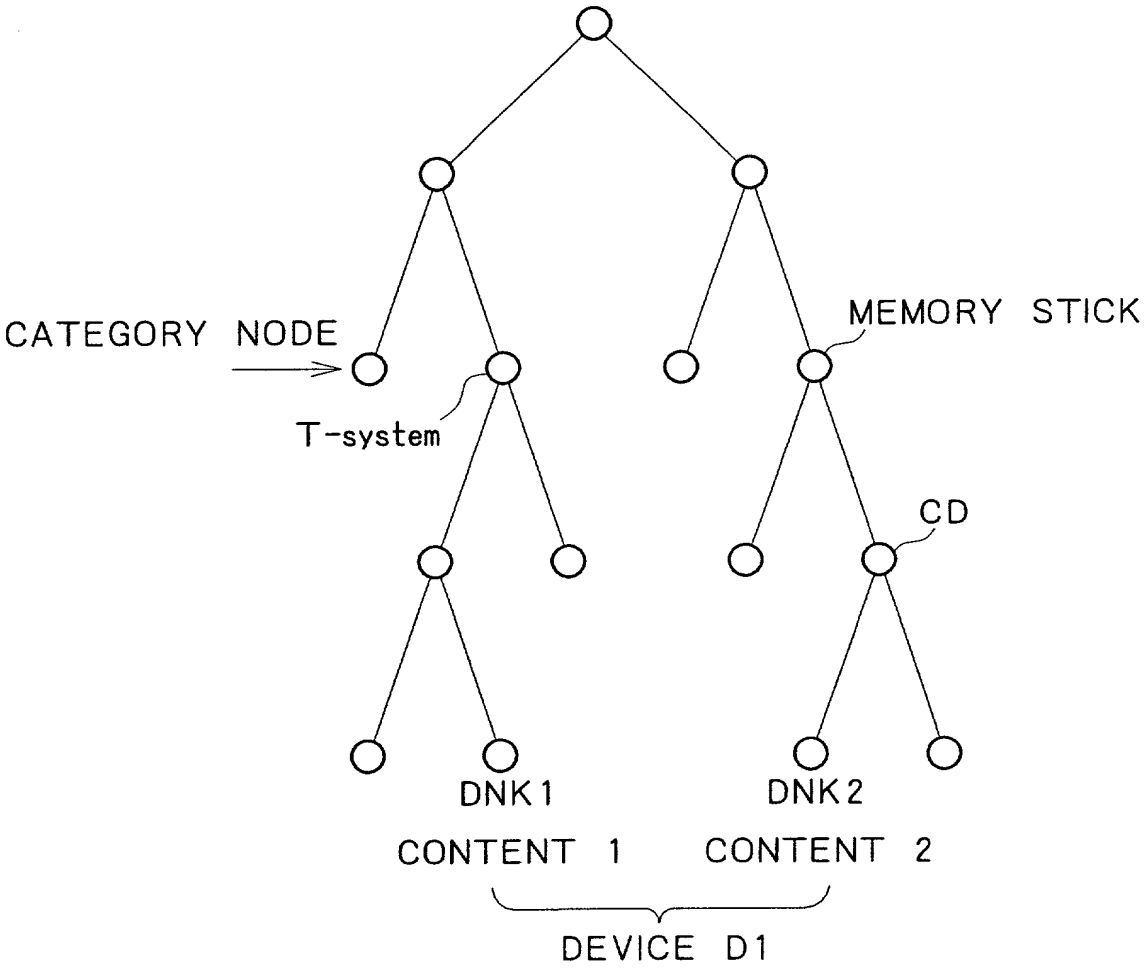

CATEGORY NODE
T-system
MEMORY STICK
CD
DNK1
DNK2
CONTENT 1
CONTENT 2
DEVICE D1

FIG. 22

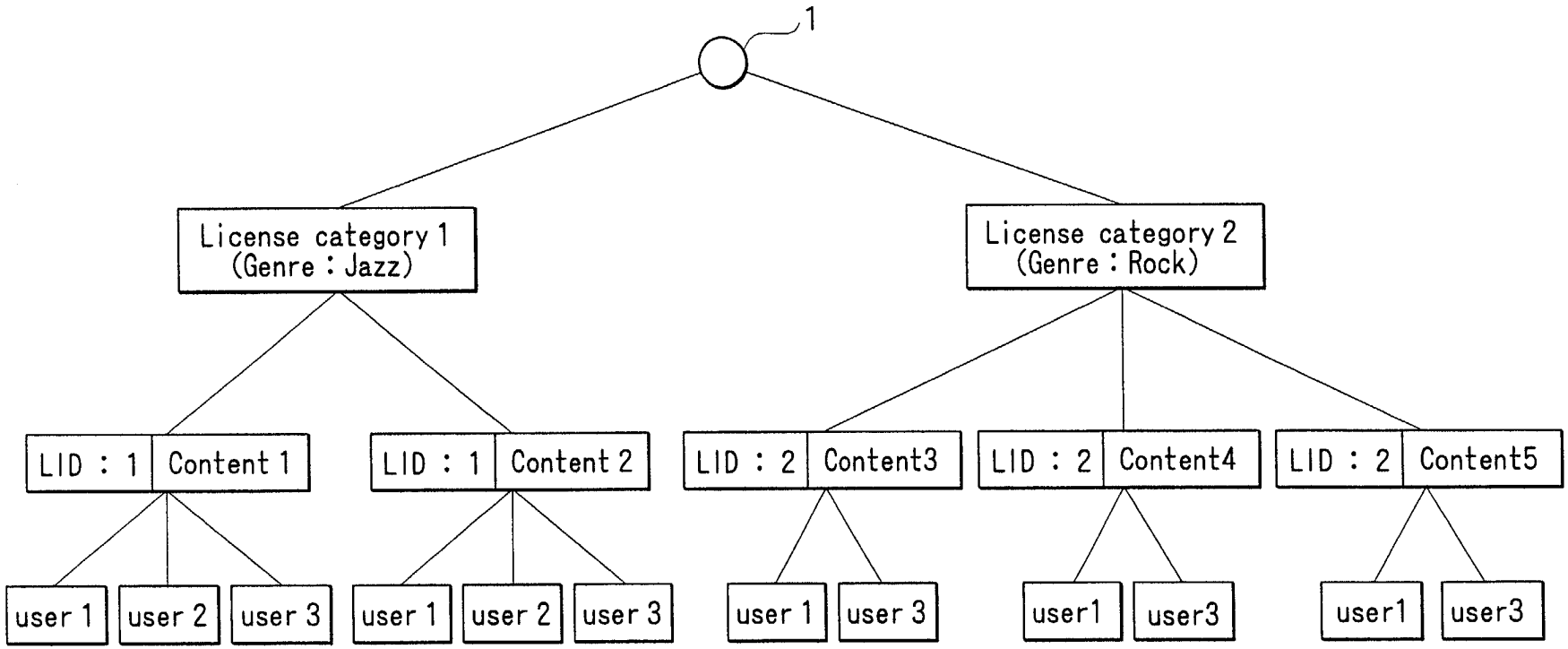

1
License category 1
(Genre : Jazz)
License category 2
(Genre : Rock)
LID : 1
Content 1
LID : 1
Content 2
LID : 2
Content3
LID : 2
Content4
LID : 2
Content5
user 1
user 2
user 3
user 1
user 2
user 3
user 1
user 3
user1
user3
user1
user3

FIG. 23

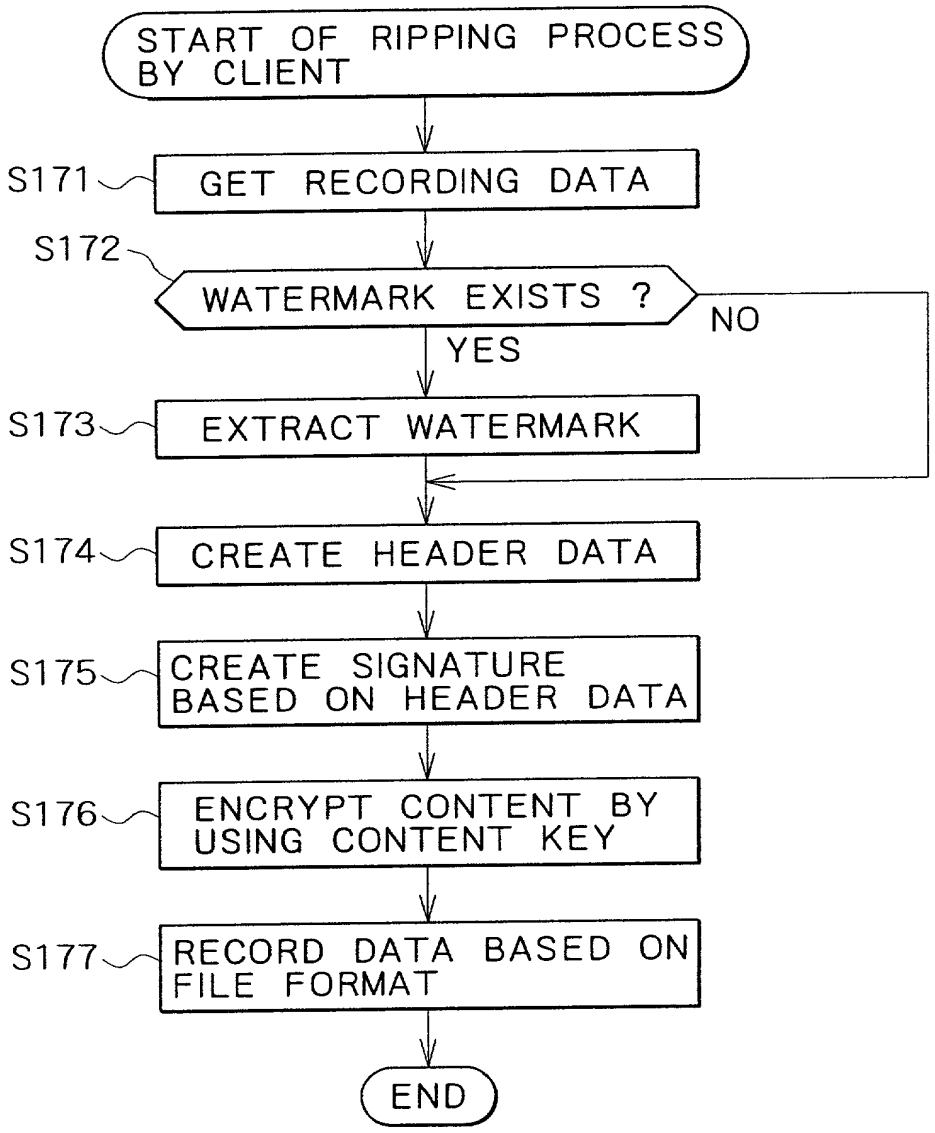

START OF RIPPING PROCESS BY CLIENT
S171
GET RECORDING DATA
S172
WATERMARK EXISTS ?
NO
YES
S173
EXTRACT WATERMARK
S174
CREATE HEADER DATA
S175
CREATE SIGNATURE BASED ON HEADER DATA
S176
ENCRYPT CONTENT BY USING CONTENT KEY
S177
RECORD DATA BASED ON FILE FORMAT
END

FIG. 24

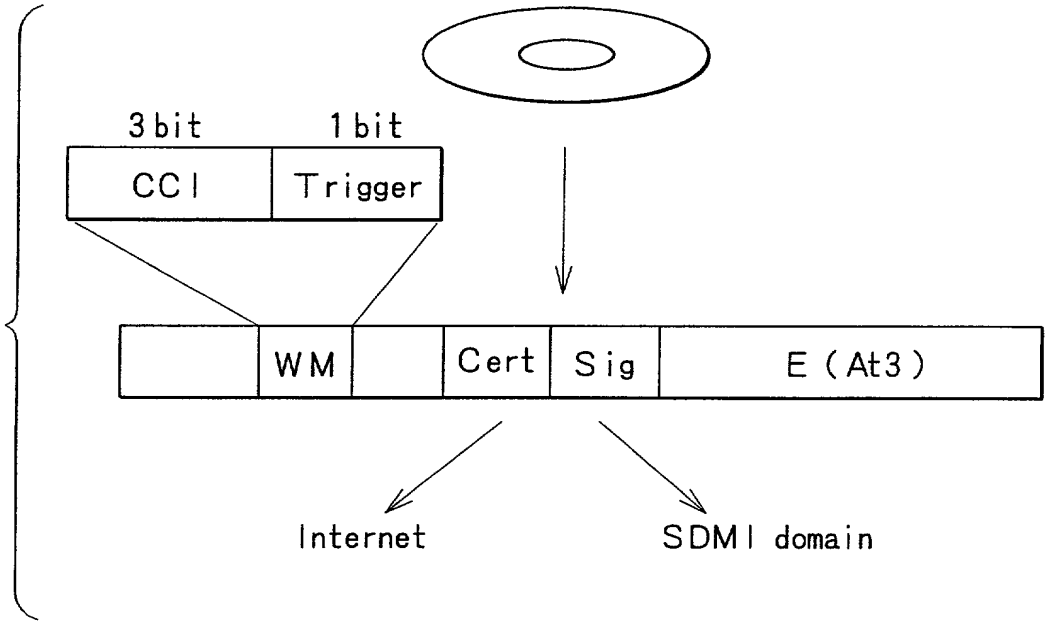

3bit
1bit
CCI
Trigger
WM
Cert
Sig
E (At3)
Internet
SDMI domain

FIG. 25

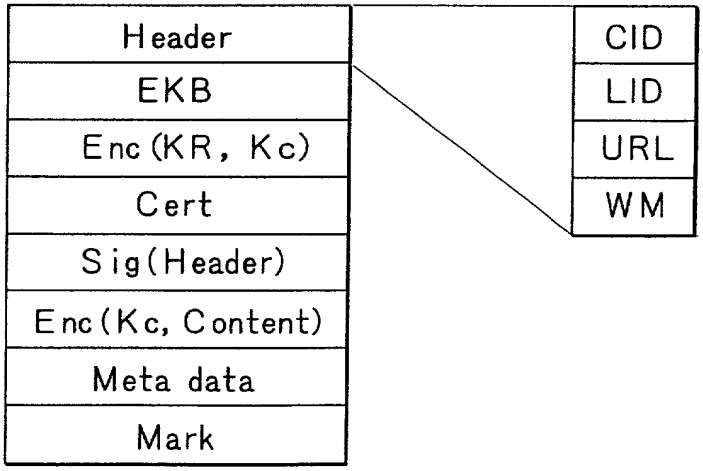

Header
EKB
Enc(KR, Kc)
Cert
Sig(Header)
Enc(Kc, Content)
Meta data
Mark
CID
LID
URL
WM

FIG. 26

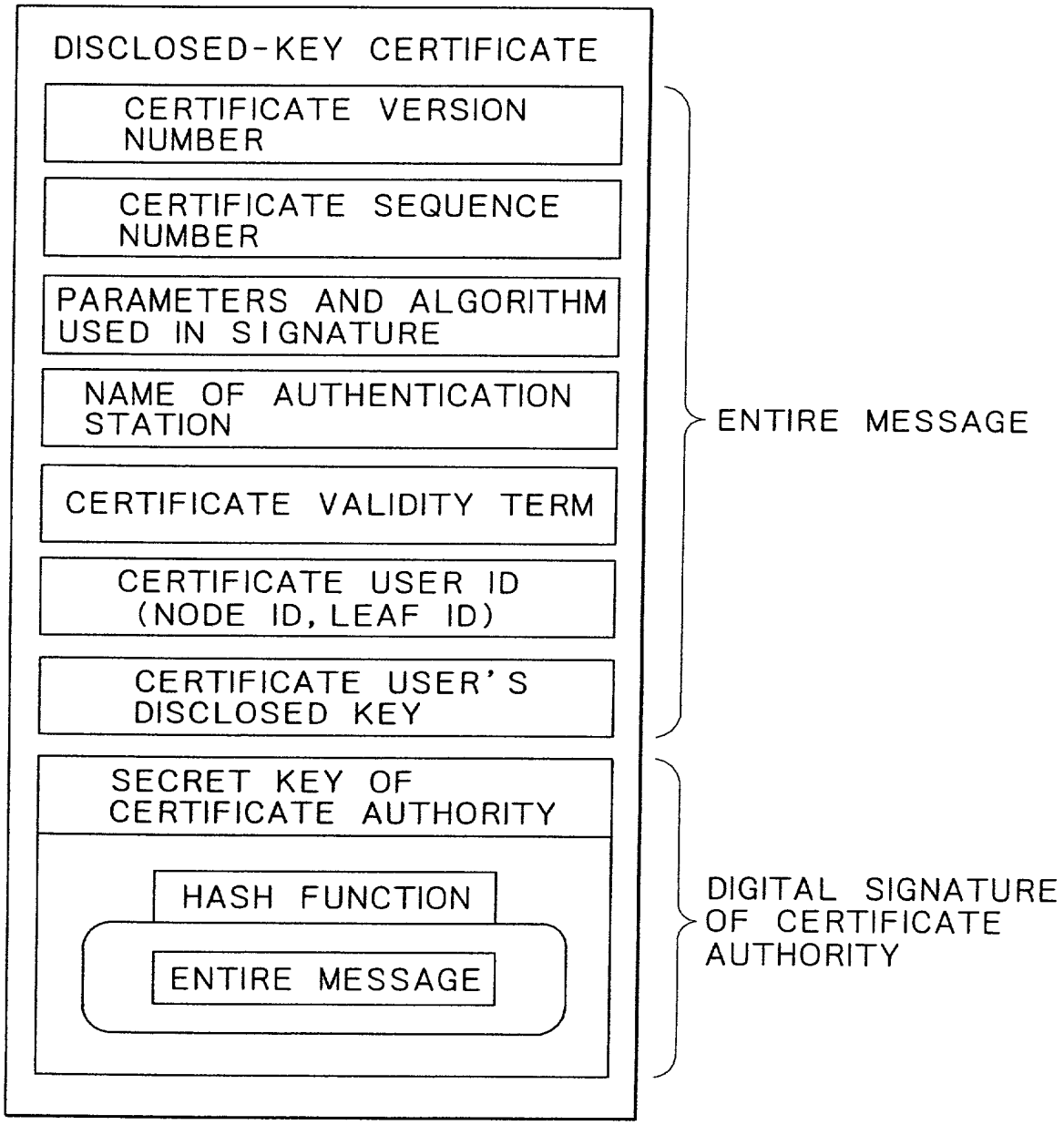

DISCLOSED-KEY CERTIFICATE
CERTIFICATE VERSION NUMBER
CERTIFICATE SEQUENCE NUMBER
PARAMETERS AND ALGORITHM USED IN SIGNATURE
NAME OF AUTHENTICATION STATION
CERTIFICATE VALIDITY TERM
CERTIFICATE USER ID (NODE ID, LEAF ID)
CERTIFICATE USER'S DISCLOSED KEY
SECRET KEY OF CERTIFICATE AUTHORITY
HASH FUNCTION
ENTIRE MESSAGE
ENTIRE MESSAGE
DIGITAL SIGNATURE OF CERTIFICATE AUTHORITY

FIG. 27

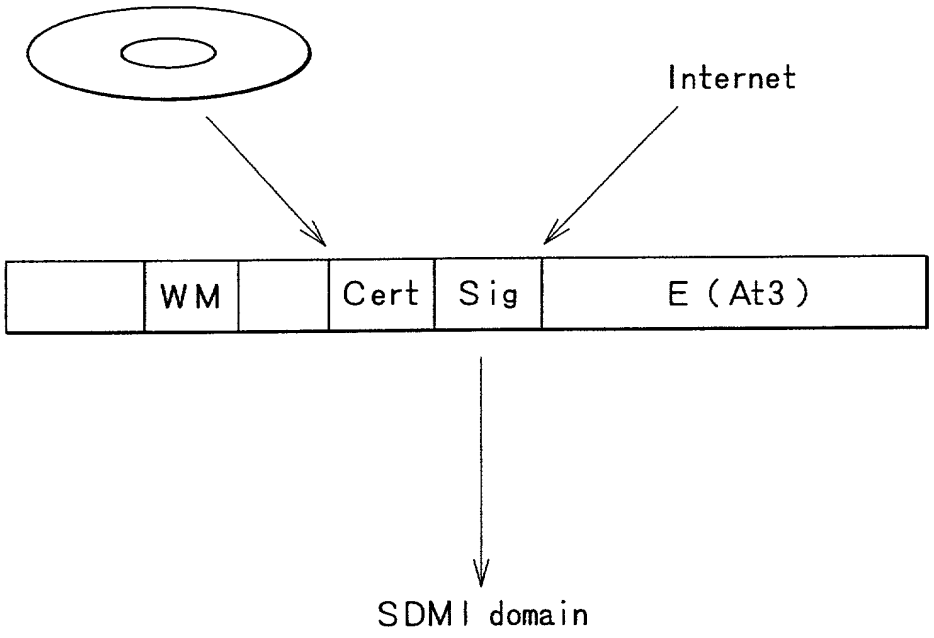


FIG. 28

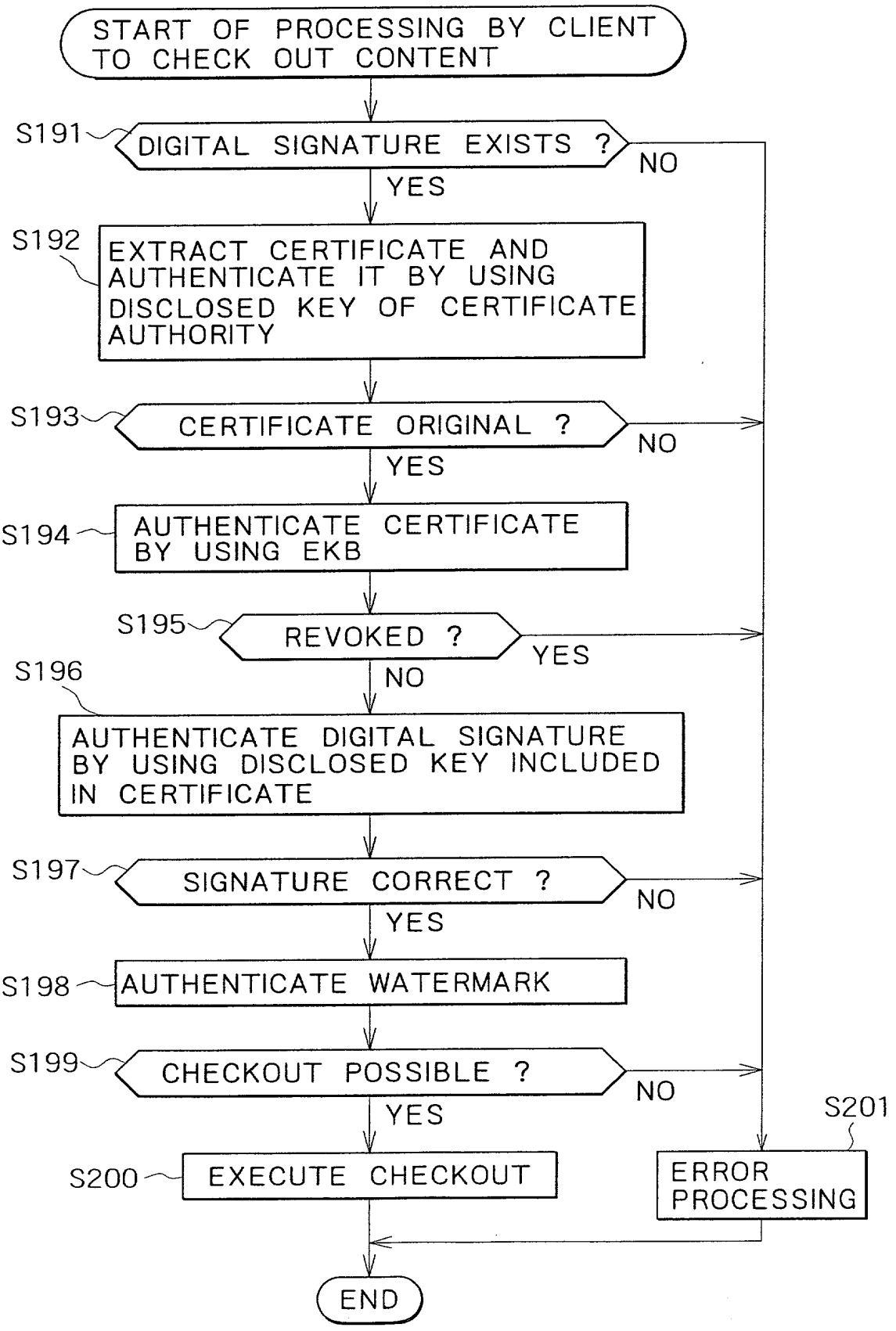

START OF PROCESSING BY CLIENT TO CHECK OUT CONTENT
S191
DIGITAL SIGNATURE EXISTS ?
NO
YES
S192
EXTRACT CERTIFICATE AND AUTHENTICATE IT BY USING DISCLOSED KEY OF CERTIFICATE AUTHORITY
S193
CERTIFICATE ORIGINAL ?
NO
YES
S194
AUTHENTICATE CERTIFICATE BY USING EKB
S195
REVOKED ?
YES
NO
S196
AUTHENTICATE DIGITAL SIGNATURE BY USING DISCLOSED KEY INCLUDED IN CERTIFICATE
S197
SIGNATURE CORRECT ?
NO
YES
S198
AUTHENTICATE WATERMARK
S199
CHECKOUT POSSIBLE ?
NO
YES
S200
EXECUTE CHECKOUT
S201
ERROR PROCESSING
END

FIG. 29

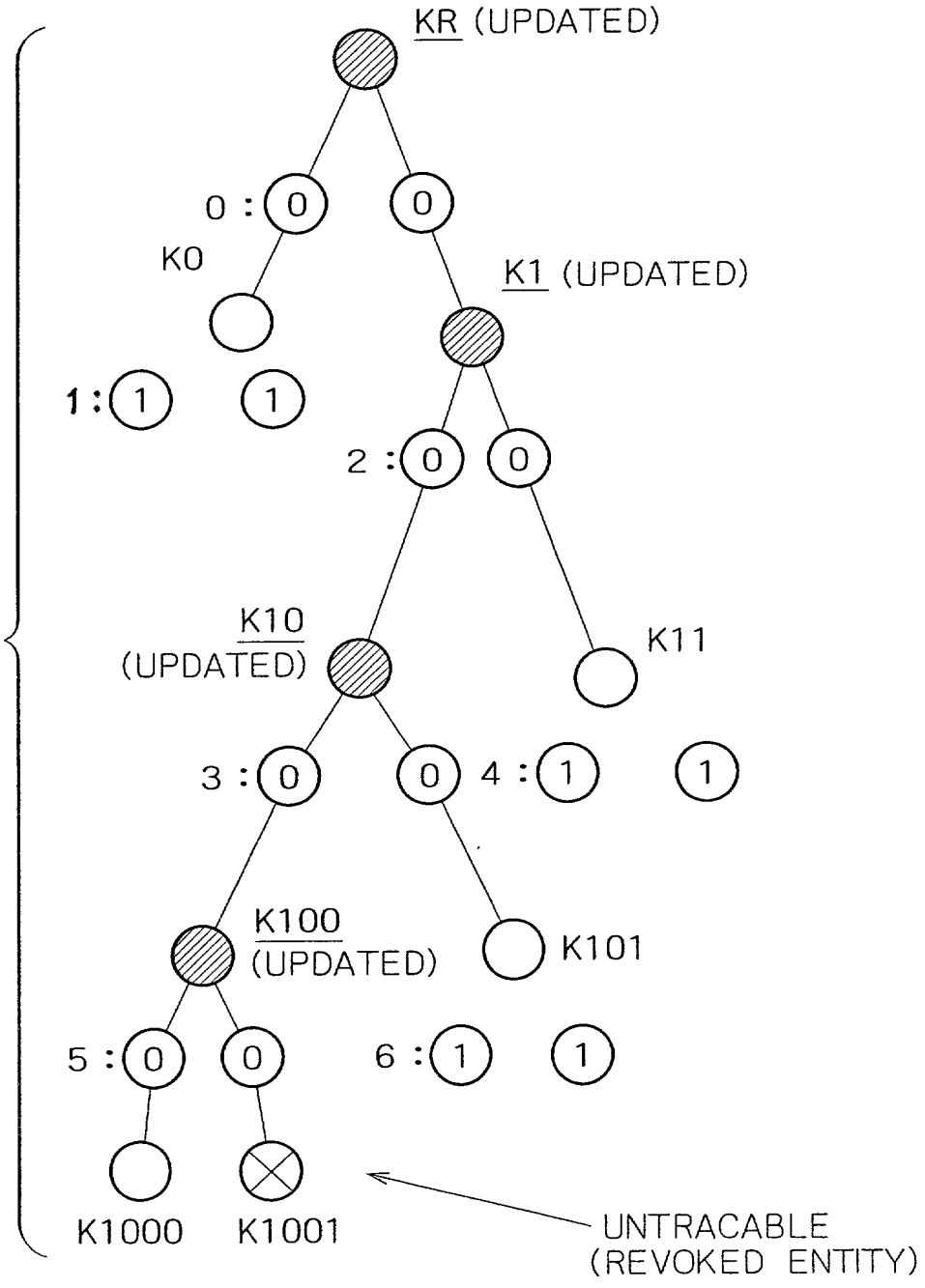

KR (UPDATED)
0 : 0
0
K0
K1 (UPDATED)
1 : 1
1
2 : 0
0
K10
(UPDATED)
K11
3 : 0
0
4 : 1
1
K100
(UPDATED)
K101
5 : 0
0
6 : 1
1
K1000
K1001
UNTRACABLE
(REVOKED ENTITY)

FIG. 30

DATA PORTION AND TAGS OF EKB
(ENABLING KEY BLOCK)

DATA (ENCRYPTED KEYS)	Enc(K0, K(t)R, Enc(K(t)1, K(t)R) Enc(K(t)10, K(t)1), Enc(K11, K(t)1) Enc(K(t)100, K(t)10), Enc(K101, K(t)10) Enc(K1000, K(t)100)
TAGS	0 : {0,0}, 1 : {1,1}, 2 : {0,0}, 3 : {0,0} 4 : {1,1}, 5 : {0,1}, 6 : {1,1}

↕

{L TAG, R TAG}

L TAG AND R TAG ARE EACH 0 TO INDICATE EXISTENCE OF DATA OR 1 OTHERWISE

FIG. 31

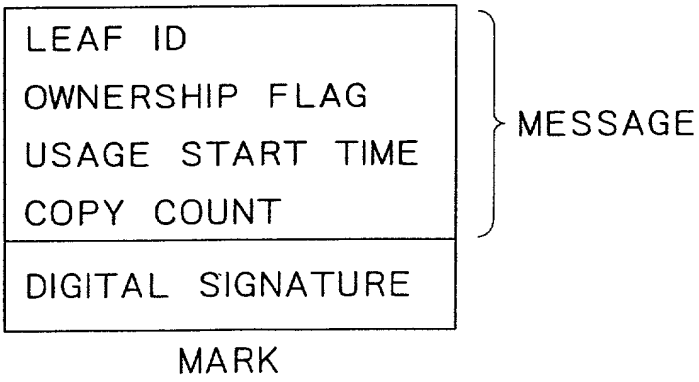


FIG. 32

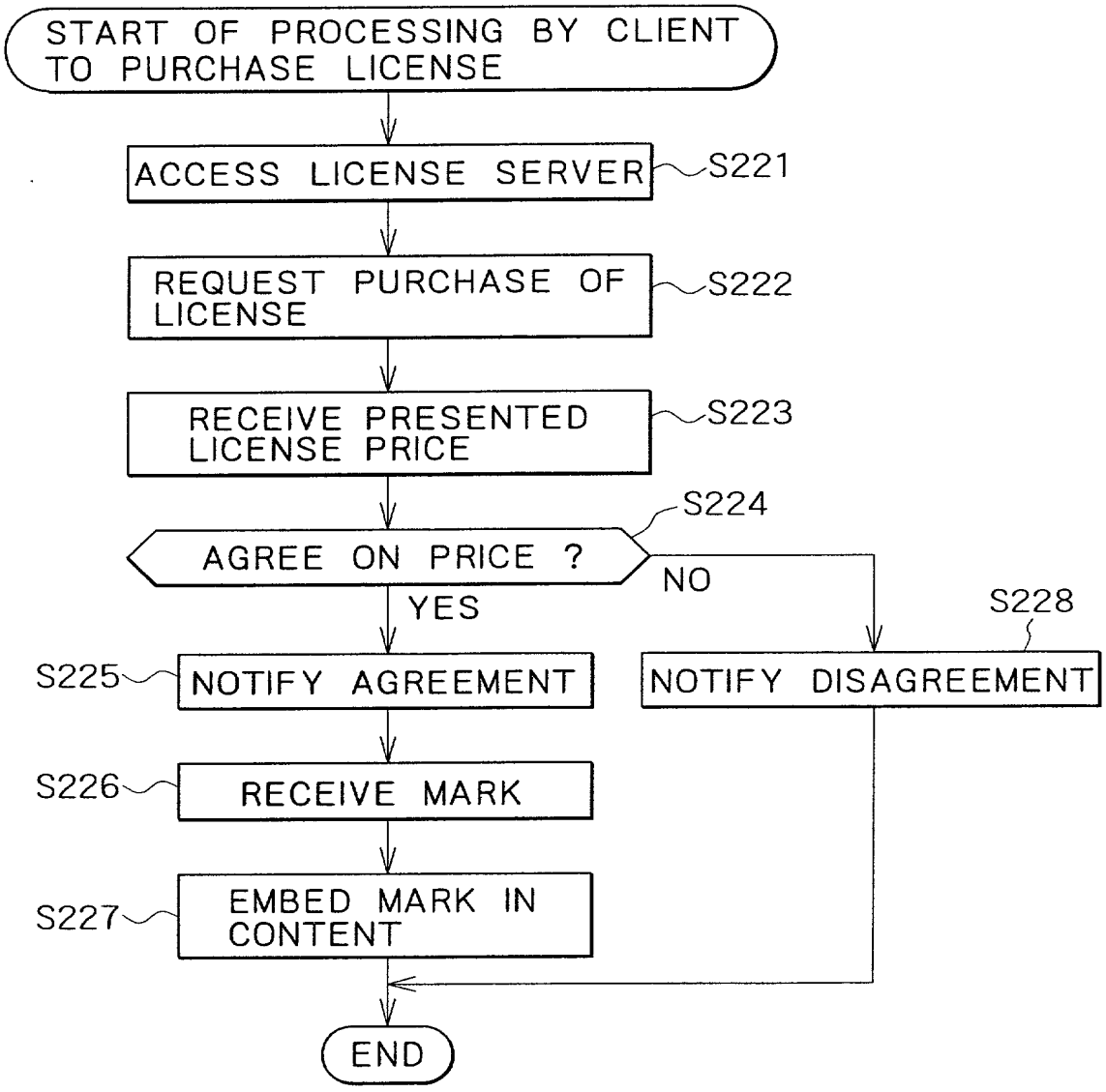

START OF PROCESSING BY CLIENT TO PURCHASE LICENSE
ACCESS LICENSE SERVER
S221
REQUEST PURCHASE OF LICENSE
S222
RECEIVE PRESENTED LICENSE PRICE
S223
S224
AGREE ON PRICE ?
NO
YES
S228
S225
NOTIFY AGREEMENT
NOTIFY DISAGREEMENT
S226
RECEIVE MARK
S227
EMBED MARK IN CONTENT
END

FIG. 33

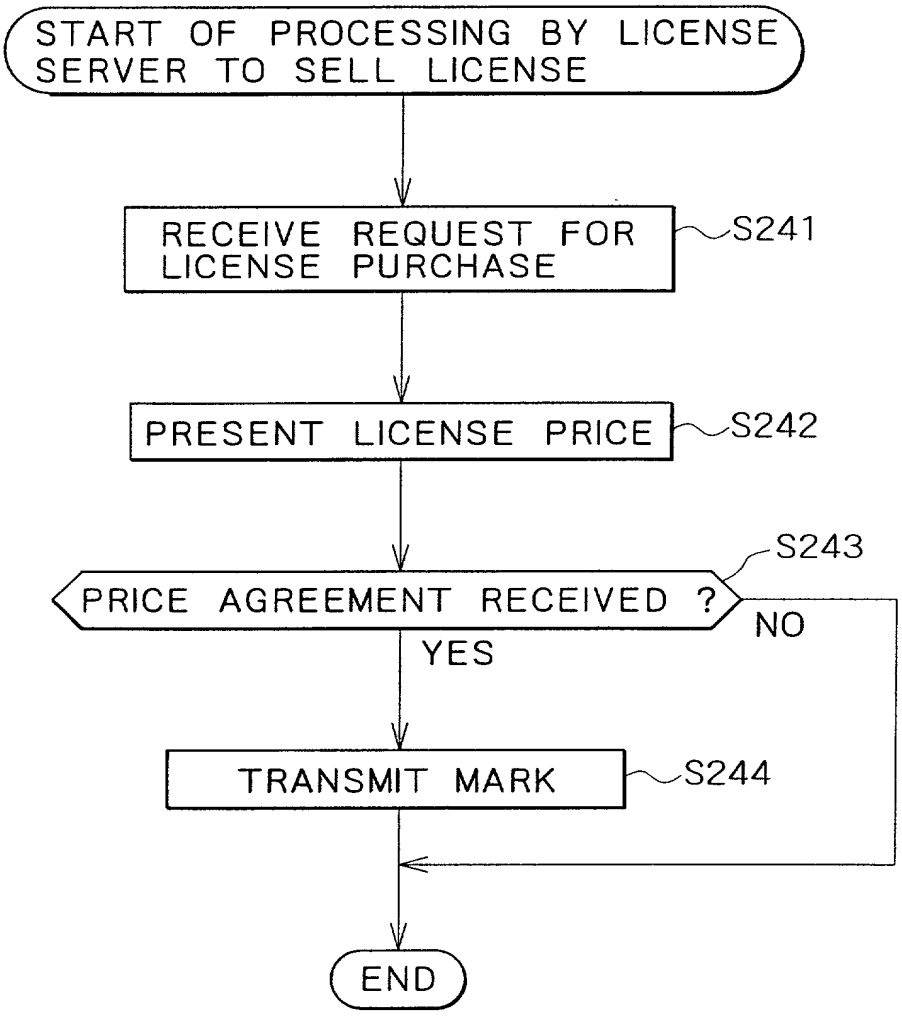

START OF PROCESSING BY LICENSE SERVER TO SELL LICENSE
RECEIVE REQUEST FOR LICENSE PURCHASE
S241
PRESENT LICENSE PRICE
S242
S243
PRICE AGREEMENT RECEIVED ?
NO
YES
TRANSMIT MARK
S244
END

FIG. 34

$$Mark = \{LeafID, Own, Sig_S(LeafID, Own)\}$$

FIG. 35

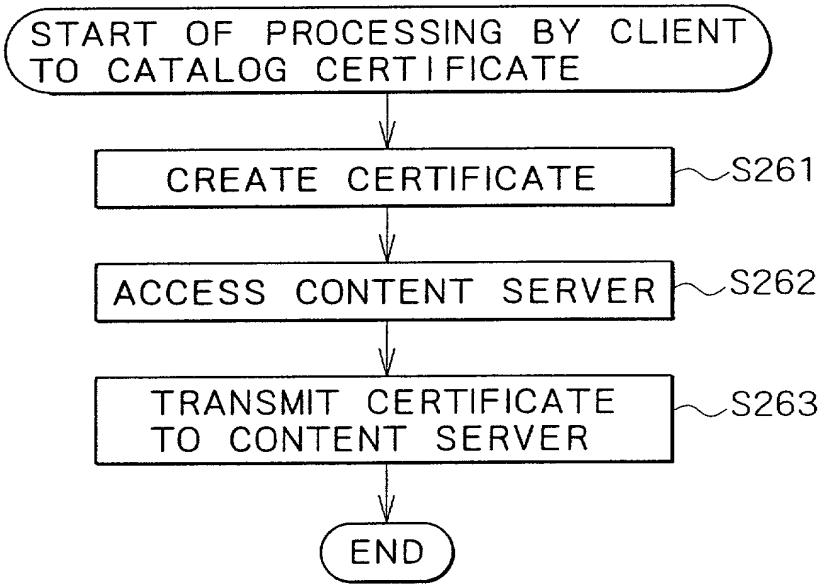


FIG. 36

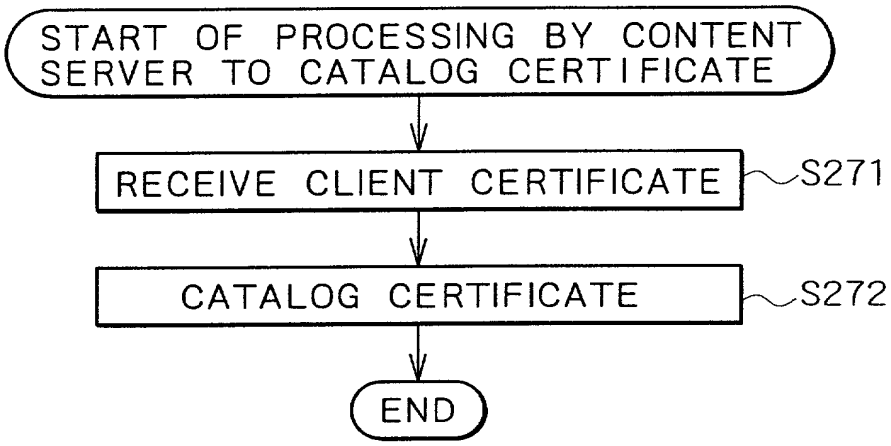

START OF PROCESSING BY CONTENT SERVER TO CATALOG CERTIFICATE
RECEIVE CLIENT CERTIFICATE
S271
CATALOG CERTIFICATE
S272
END

FIG. 37

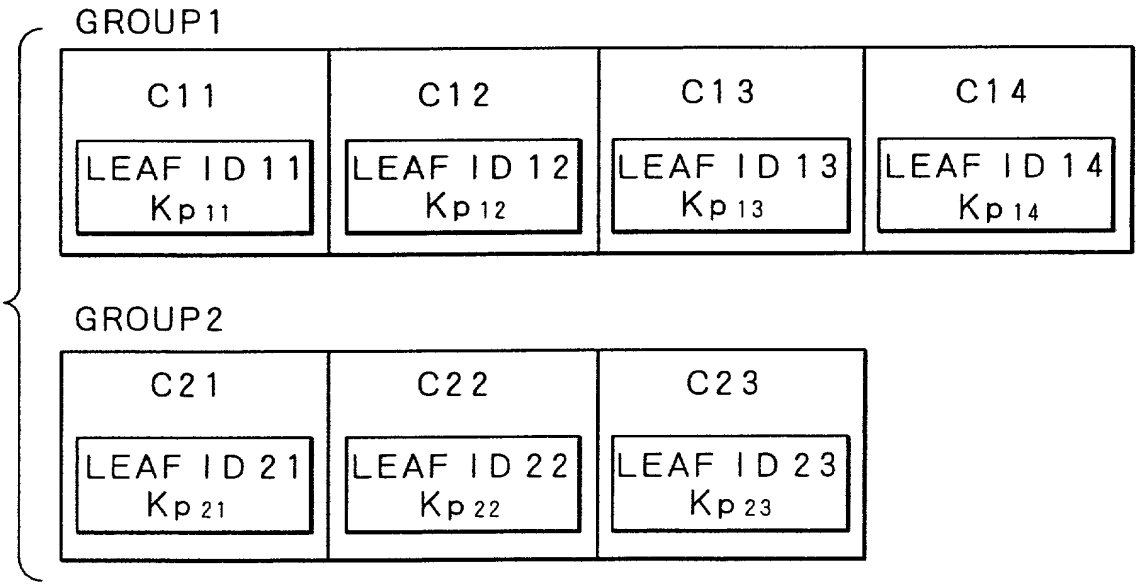

GROUP1
C11
LEAF ID 11
Kp 11
C12
LEAF ID 12
Kp 12
C13
LEAF ID 13
Kp 13
C14
LEAF ID 14
Kp 14
GROUP2
C21
LEAF ID 21
Kp 21
C22
LEAF ID 22
Kp 22
C23
LEAF ID 23
Kp 23

FIG. 38

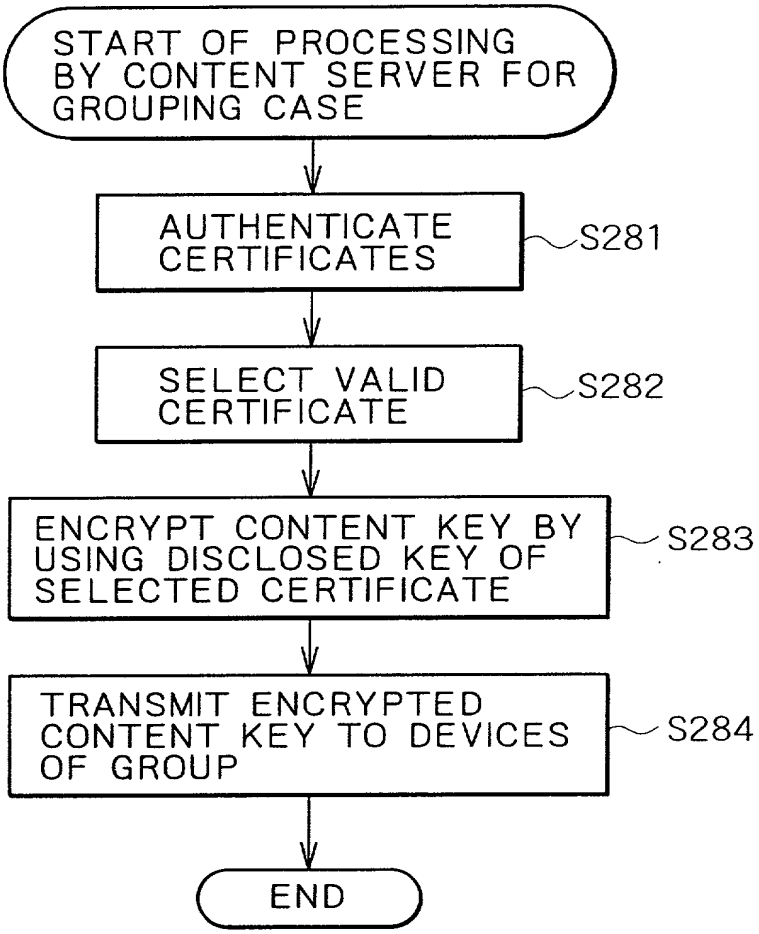


FIG. 39

$Enc(Kp_{11}, Kc), Enc(Kp_{12}, Kc), Enc(Kp_{13}, Kc)$

FIG. 40

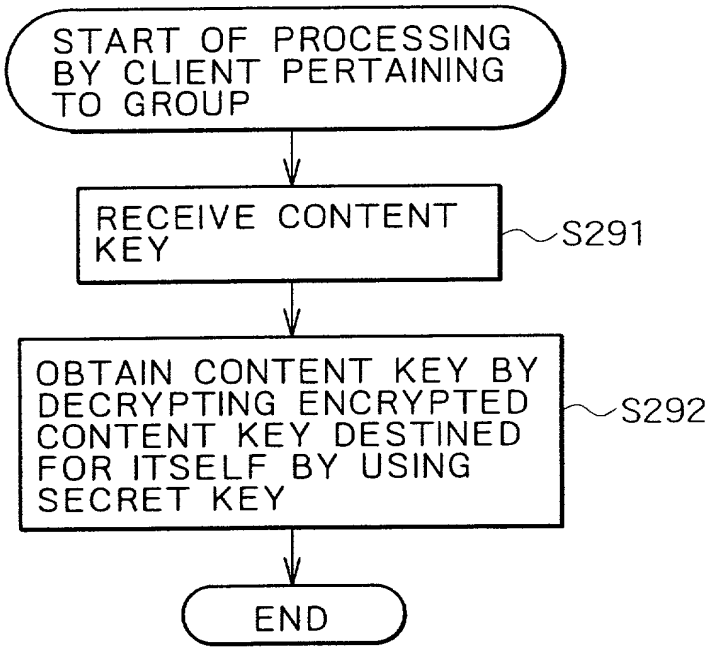

START OF PROCESSING BY CLIENT PERTAINING TO GROUP
RECEIVE CONTENT KEY
S291
OBTAIN CONTENT KEY BY DECRYPTING ENCRYPTED CONTENT KEY DESTINED FOR ITSELF BY USING SECRET KEY
S292
END

FIG. 41

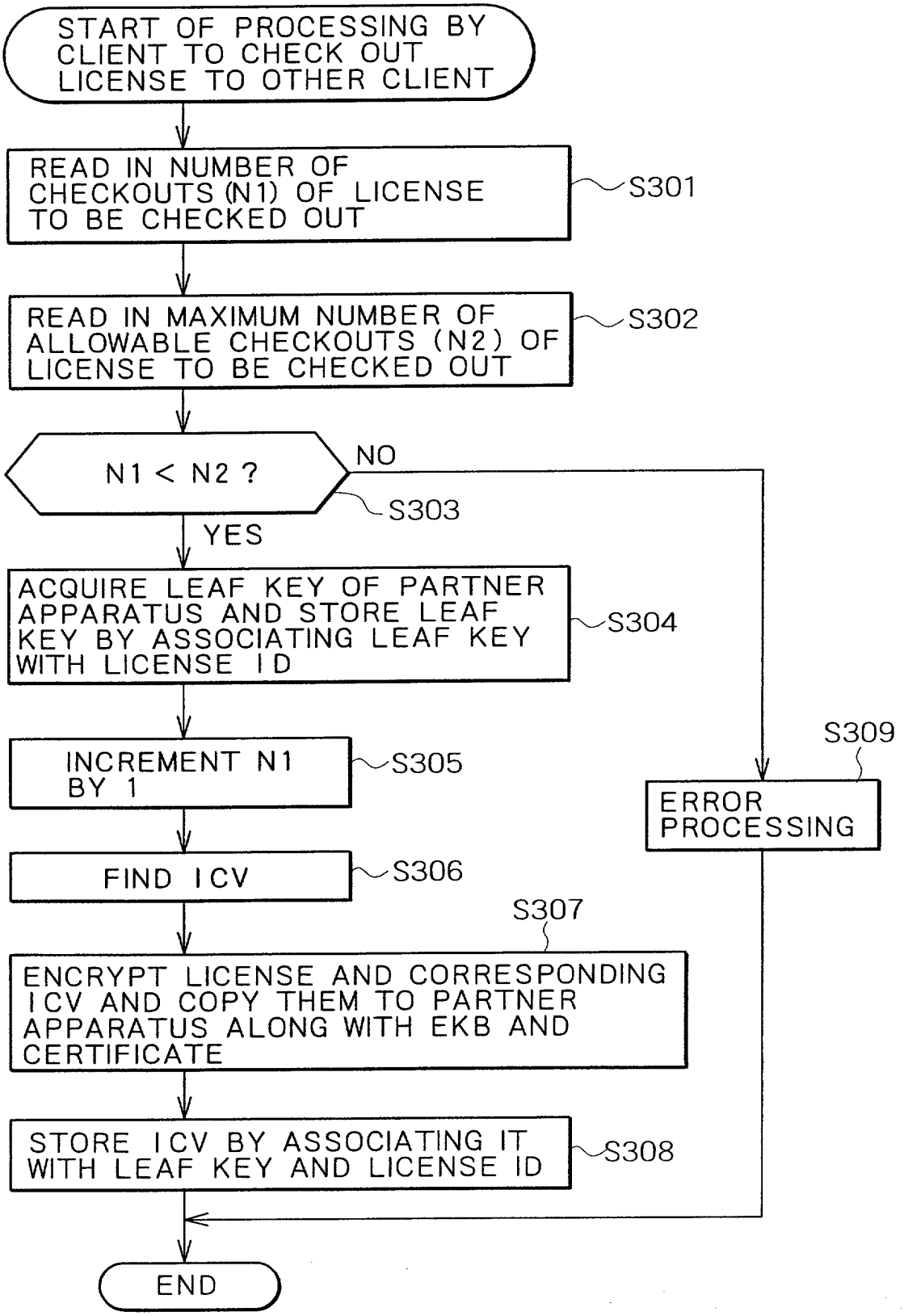

START OF PROCESSING BY CLIENT TO CHECK OUT LICENSE TO OTHER CLIENT
READ IN NUMBER OF CHECKOUTS (N1) OF LICENSE TO BE CHECKED OUT
S301
READ IN MAXIMUM NUMBER OF ALLOWABLE CHECKOUTS (N2) OF LICENSE TO BE CHECKED OUT
S302
N1 < N2 ?
NO
S303
YES
ACQUIRE LEAF KEY OF PARTNER APPARATUS AND STORE LEAF KEY BY ASSOCIATING LEAF KEY WITH LICENSE ID
S304
INCREMENT N1 BY 1
S305
FIND ICV
S306
S307
ENCRYPT LICENSE AND CORRESPONDING ICV AND COPY THEM TO PARTNER APPARATUS ALONG WITH EKB AND CERTIFICATE
STORE ICV BY ASSOCIATING IT WITH LEAF KEY AND LICENSE ID
S308
S309
ERROR PROCESSING
END

FIG. 42

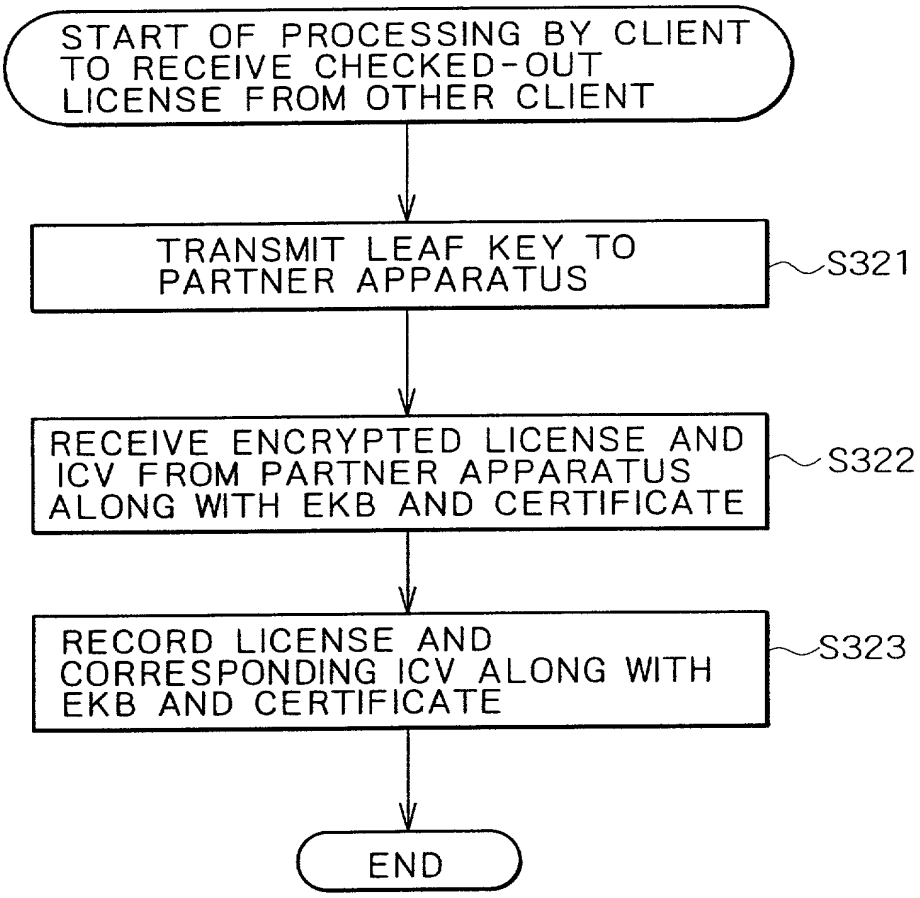

START OF PROCESSING BY CLIENT TO RECEIVE CHECKED-OUT LICENSE FROM OTHER CLIENT
TRANSMIT LEAF KEY TO PARTNER APPARATUS
S321
RECEIVE ENCRYPTED LICENSE AND ICV FROM PARTNER APPARATUS ALONG WITH EKB AND CERTIFICATE
S322
RECORD LICENSE AND CORRESPONDING ICV ALONG WITH EKB AND CERTIFICATE
S323
END

FIG. 43

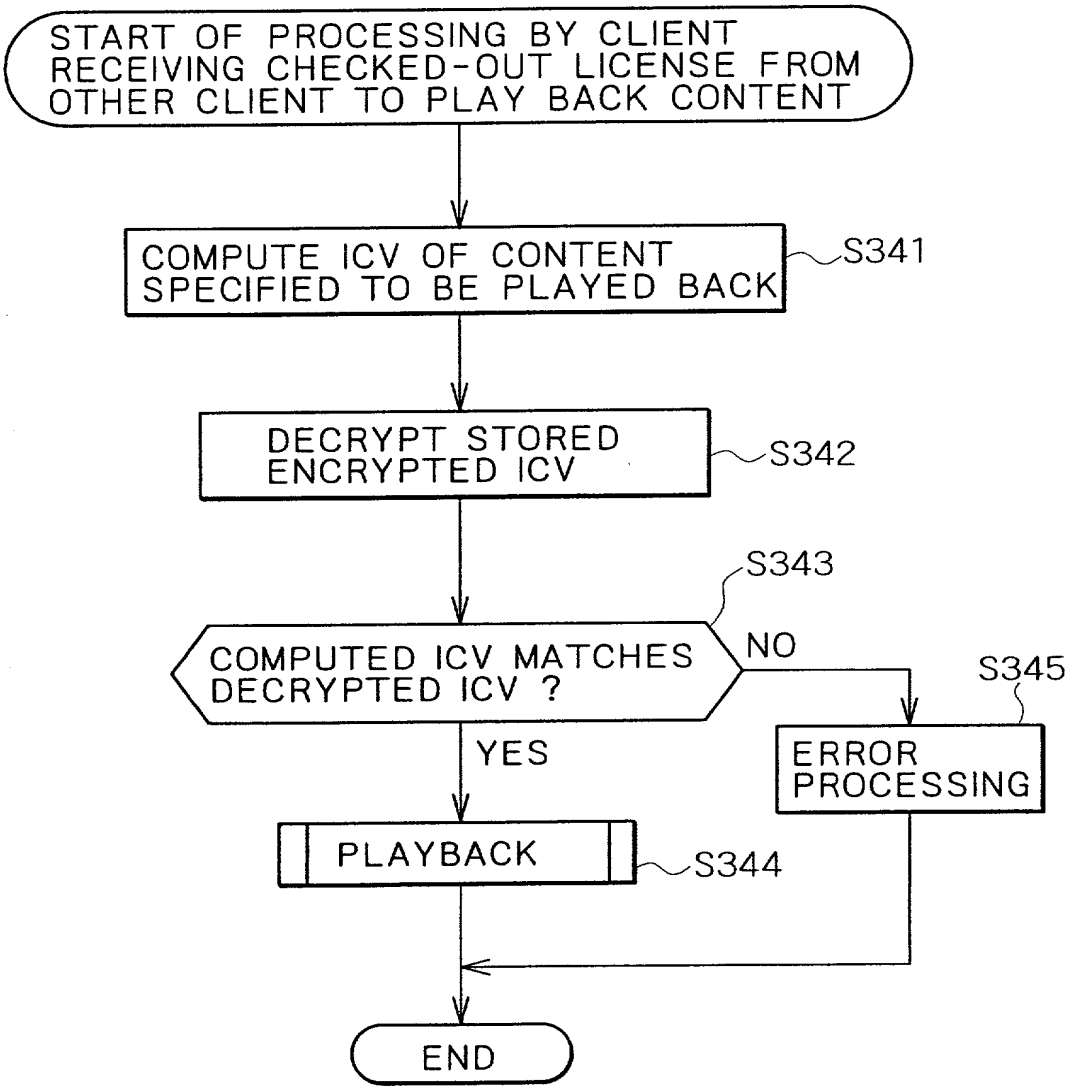

START OF PROCESSING BY CLIENT RECEIVING CHECKED-OUT LICENSE FROM OTHER CLIENT TO PLAY BACK CONTENT
COMPUTE ICV OF CONTENT SPECIFIED TO BE PLAYED BACK
S341
DECRYPT STORED ENCRYPTED ICV
S342
S343
COMPUTED ICV MATCHES DECRYPTED ICV ?
NO
S345
ERROR PROCESSING
YES
PLAYBACK
S344
END

FIG. 44

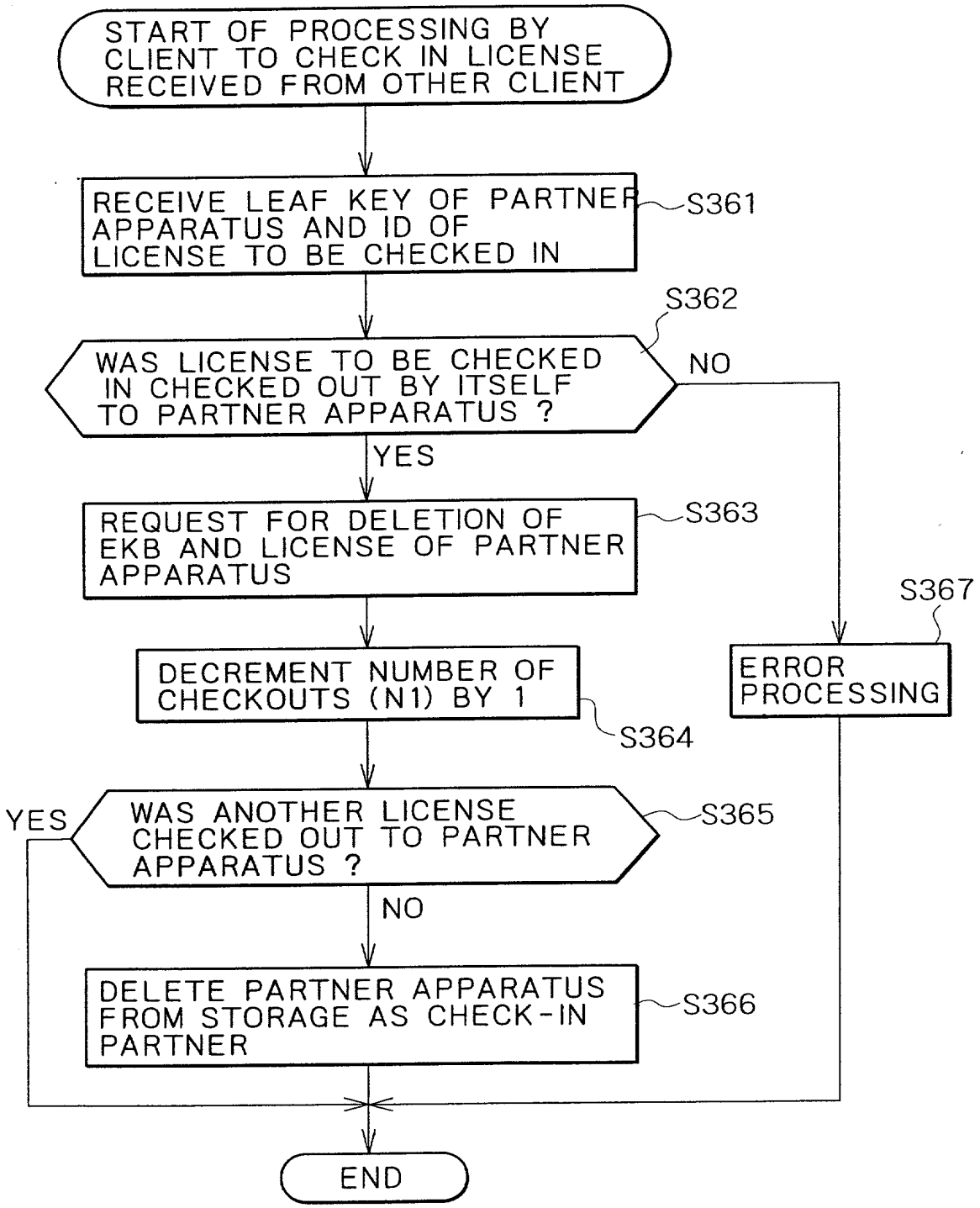

START OF PROCESSING BY CLIENT TO CHECK IN LICENSE RECEIVED FROM OTHER CLIENT
RECEIVE LEAF KEY OF PARTNER APPARATUS AND ID OF LICENSE TO BE CHECKED IN
S361
S362
WAS LICENSE TO BE CHECKED IN CHECKED OUT BY ITSELF TO PARTNER APPARATUS ?
NO
YES
REQUEST FOR DELETION OF EKB AND LICENSE OF PARTNER APPARATUS
S363
S367
ERROR PROCESSING
DECREMENT NUMBER OF CHECKOUTS (N1) BY 1
S364
YES
WAS ANOTHER LICENSE CHECKED OUT TO PARTNER APPARATUS ?
S365
NO
DELETE PARTNER APPARATUS FROM STORAGE AS CHECK-IN PARTNER
S366
END

FIG. 45

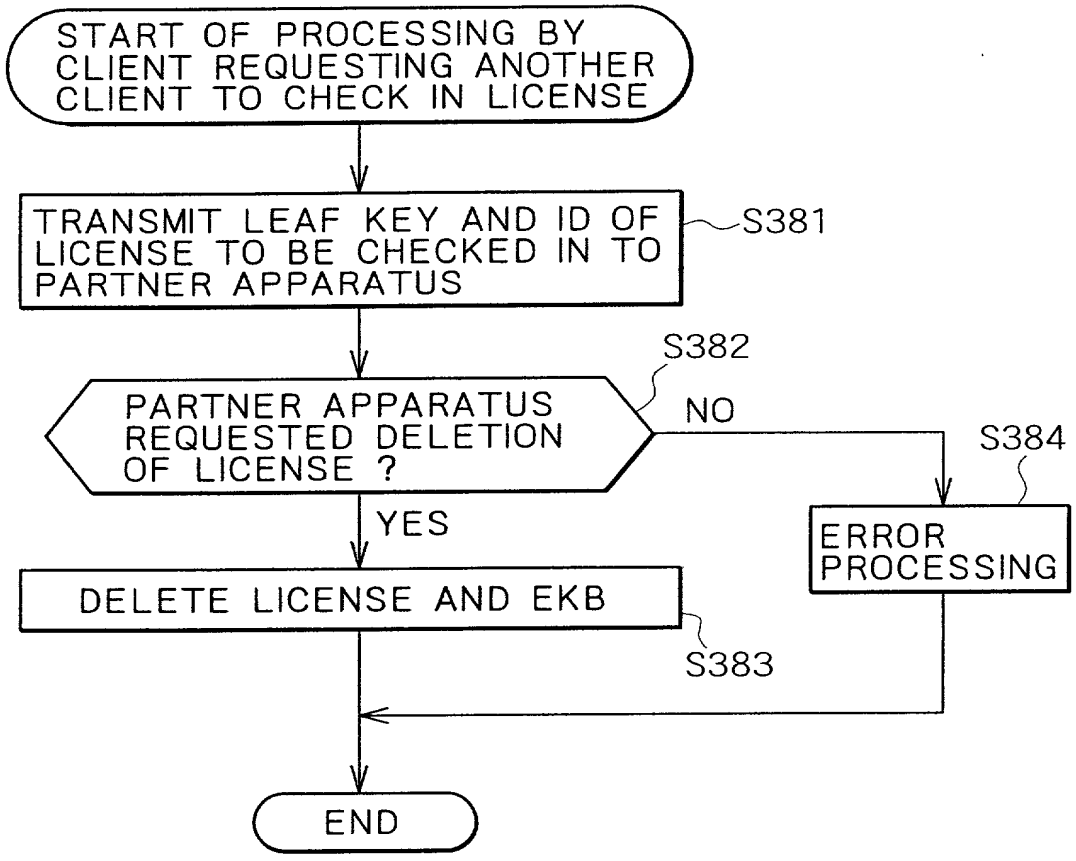

START OF PROCESSING BY CLIENT REQUESTING ANOTHER CLIENT TO CHECK IN LICENSE
TRANSMIT LEAF KEY AND ID OF LICENSE TO BE CHECKED IN TO PARTNER APPARATUS
S381
S382
PARTNER APPARATUS REQUESTED DELETION OF LICENSE ?
NO
S384
YES
ERROR PROCESSING
DELETE LICENSE AND EKB
S383
END

FIG. 46

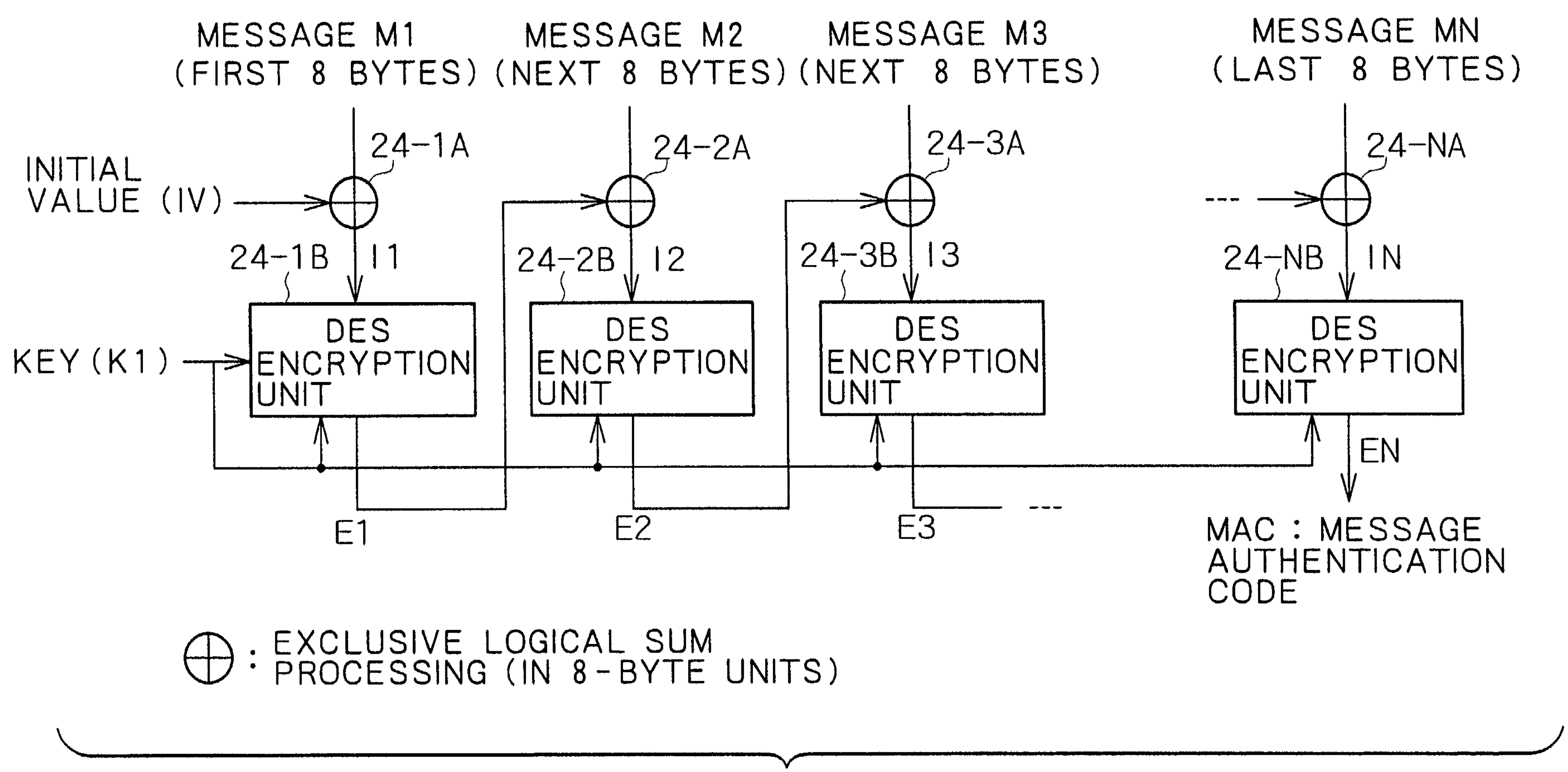

MESSAGE M1
(FIRST 8 BYTES)
MESSAGE M2
(NEXT 8 BYTES)
MESSAGE M3
(NEXT 8 BYTES)
MESSAGE MN
(LAST 8 BYTES)
INITIAL VALUE (IV)
KEY (K1)
24-1A
24-2A
24-3A
24-NA
24-1B
24-2B
24-3B
24-NB
I1
I2
I3
IN
DES ENCRYPTION UNIT
DES ENCRYPTION UNIT
DES ENCRYPTION UNIT
DES ENCRYPTION UNIT
E1
E2
E3
EN
MAC : MESSAGE AUTHENTICATION CODE
⊕ : EXCLUSIVE LOGICAL SUM PROCESSING (IN 8-BYTE UNITS)

FIG. 47

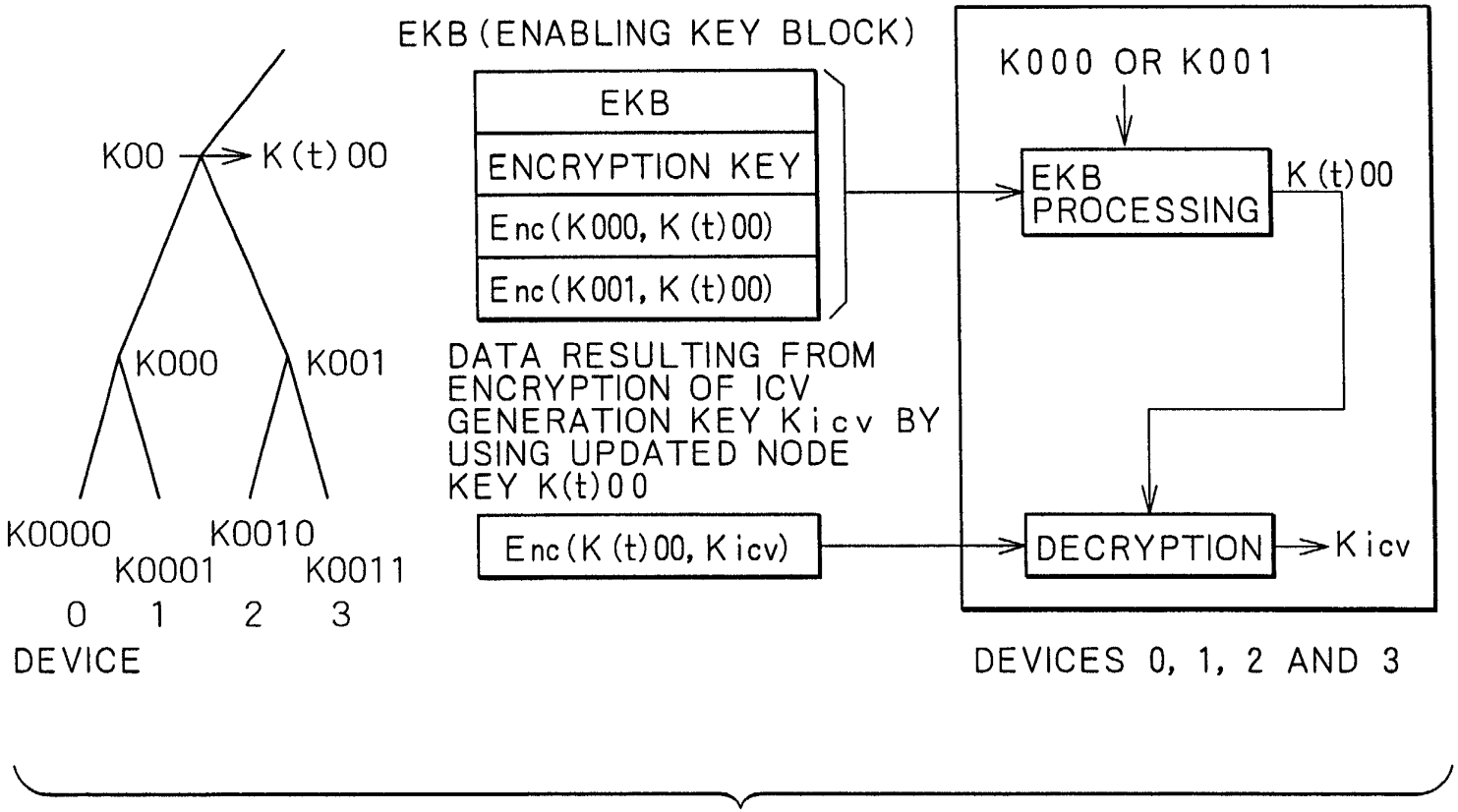

EKB (ENABLING KEY BLOCK)
EKB
ENCRYPTION KEY
Enc(K000, K(t)00)
Enc(K001, K(t)00)
K00
K(t)00
K000
K001
K0000
K0001
K0010
K0011
0
1
2
3
DEVICE
DATA RESULTING FROM ENCRYPTION OF ICV GENERATION KEY Kicv BY USING UPDATED NODE KEY K(t)00
Enc(K(t)00, Kicv)
K000 OR K001
EKB PROCESSING
K(t)00
DECRYPTION
Kicv
DEVICES 0, 1, 2 AND 3

FIG. 48

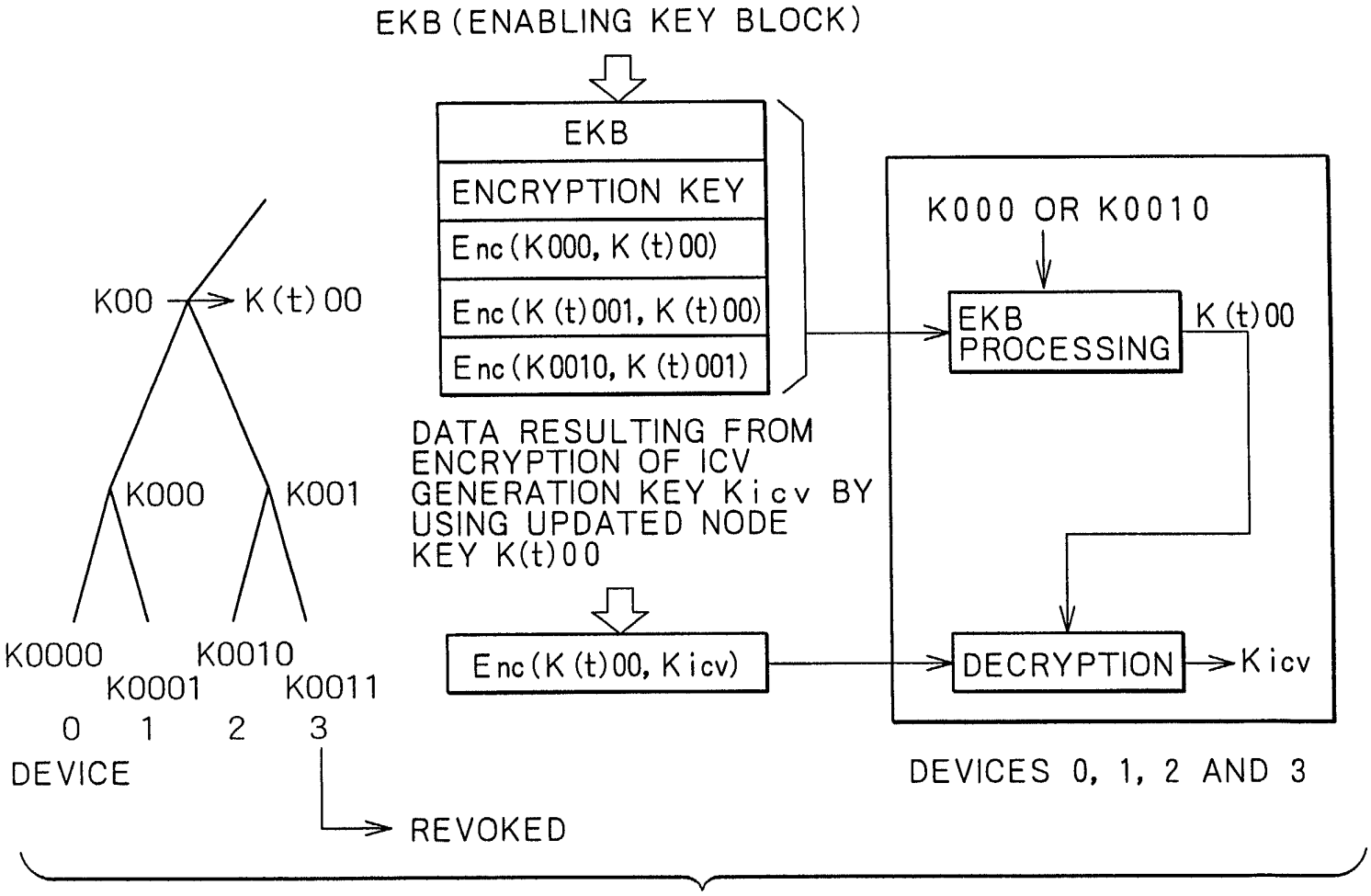

EKB (ENABLING KEY BLOCK)
EKB
ENCRYPTION KEY
Enc(K000, K(t)00)
Enc(K(t)001, K(t)00)
Enc(K0010, K(t)001)
DATA RESULTING FROM ENCRYPTION OF ICV GENERATION KEY Kicv BY USING UPDATED NODE KEY K(t)00
Enc(K(t)00, Kicv)
K00
K(t)00
K000
K001
K0000
K0001
K0010
K0011
0
1
2
3
DEVICE
REVOKED
K000 OR K0010
EKB PROCESSING
K(t)00
DECRYPTION
Kicv
DEVICES 0, 1, 2 AND 3

FIG. 49A

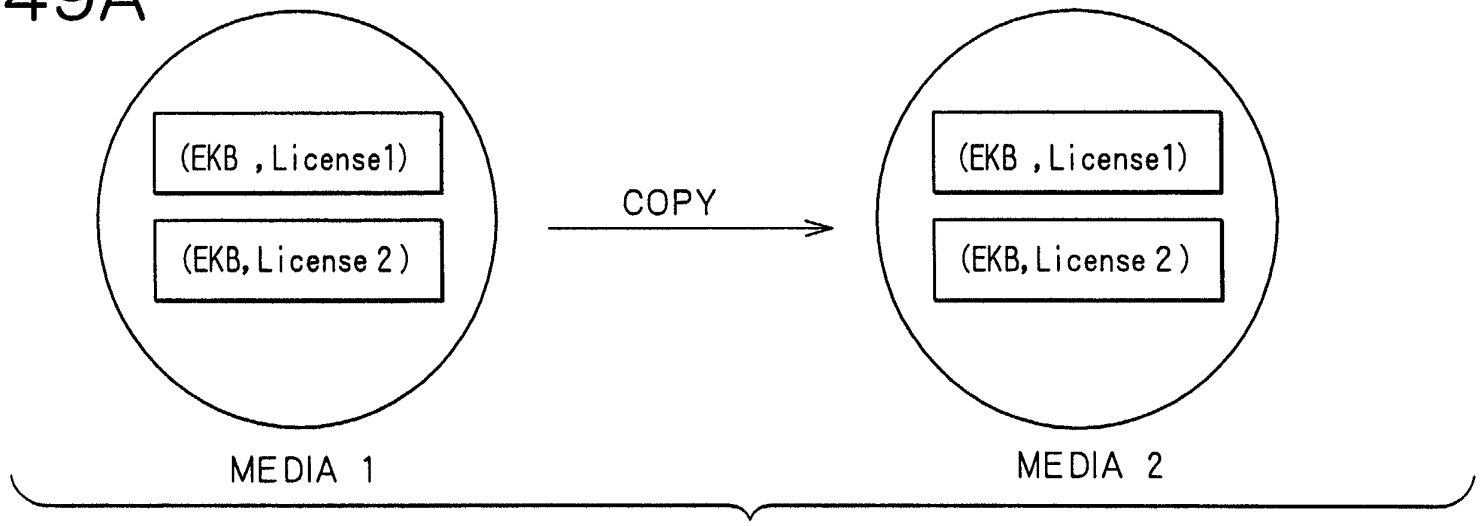


FIG. 49B

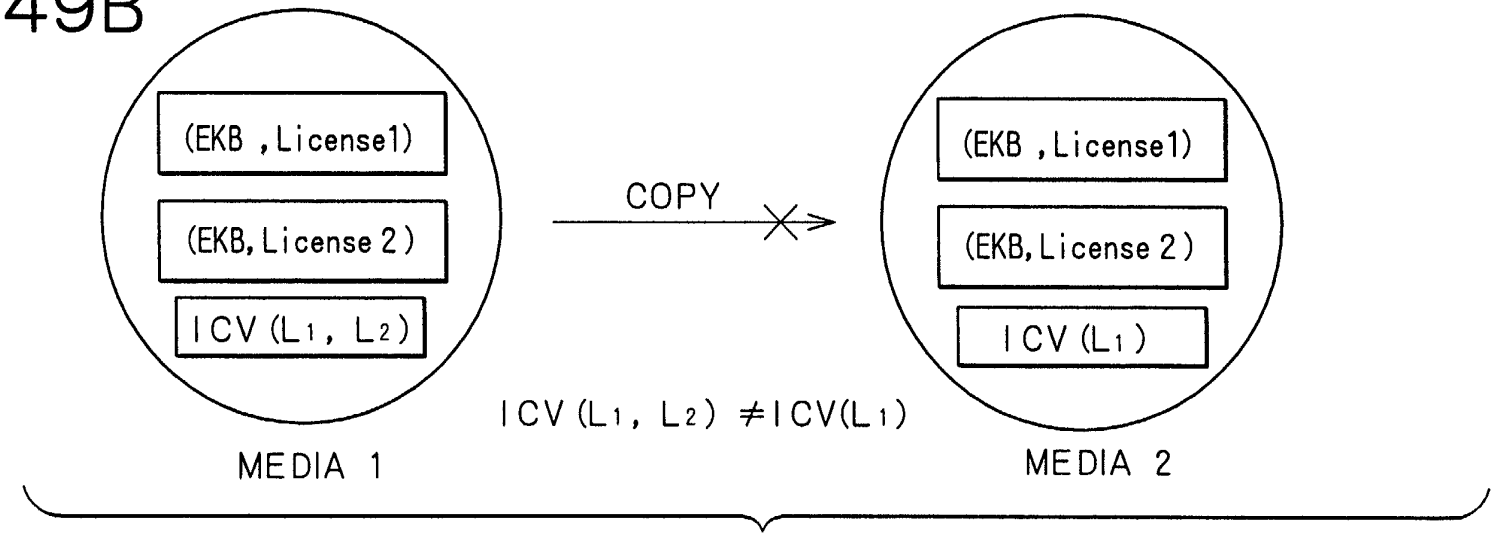
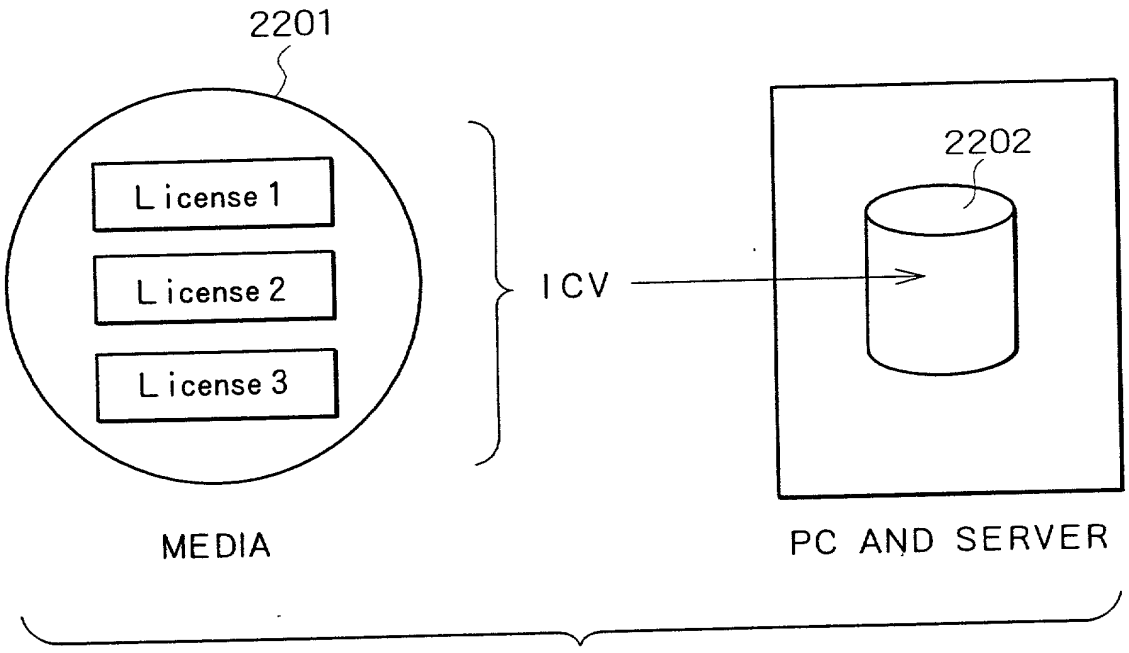


FIG. 50


2201
License 1
License 2
License 3
ICV
2202
MEDIA
PC AND SERVER